

Belief Functions: Theory and Algorithms

Thomas Reineking

February 2014

Dissertation zur Erlangung des Doktorgrades der
Ingenieurwissenschaften im Fachbereich 3 (Mathematik &
Informatik) der Universität Bremen

Date of submission: February 5, 2014

Date of defense: March 24, 2014

Dean: Prof. Dr. Kerstin Schill

Reviewers: Prof. Dr. Kerstin Schill (Universität Bremen)
Prof. Dr. Günther Palm (Universität Ulm)

Abstract

The subject of this thesis is belief function theory and its application in different contexts. Belief function theory (also known as Dempster-Shafer theory) is a mathematical framework for describing quantified beliefs held by an agent. It can be interpreted as a generalization of Bayesian probability theory and makes it possible to distinguish between different types of uncertainty. In particular, belief functions can make uncertainty resulting from a lack of evidence explicit. In this thesis, applications of belief function theory are explored both on a theoretical and on an algorithmic level. One of the major criticisms raised against the use of belief functions is the exponential complexity associated with their representation and combination. This criticism is addressed in this thesis by showing how efficient algorithms can be developed based on Monte-Carlo approximations and exploitation of independence.

First, the context of temporal processes subject to uncertainty is considered where uncertainty can be modeled by belief functions. Here, evidential temporal update equations are derived that generalize Bayesian filtering and allow the state of a dynamical system to be estimated recursively over time. In order to reduce the exponential complexity of solving these equations, a Monte-Carlo approximation resulting in an evidential particle filter algorithm is presented. This evidential particle filter algorithm constitutes a generalization of probabilistic particle filters in discrete domains and reduces the exponential time and space complexity of the analytical filtering solution to a complexity that is linear in the size of the state space.

The second context considered in this thesis is spatial uncertainty; specifically, the problem of simultaneous localization and mapping (SLAM). For SLAM, a mobile robot exploring an unknown environment is tasked with constructing a map of the environment while, at the same time, localizing itself using this map. It is proved in this thesis that the joint distribution of the robot's path and the map can be factorized into a probabilistic path component and an evidential map component. This factorized joint distribution is then approximated using a Rao-Blackwellized particle filter, resulting in an evidential SLAM algorithm that generalizes the popular probabilistic FastSLAM algorithm. The grid maps produced by the algorithm are described by belief functions and thus provide the robot with additional information about the uncertainty in the map. The time complexity of incorporating a new measurement is linear in the number of grid cells and is therefore identical to the complexity

of the probabilistic FastSLAM algorithm.

The final context explored in this thesis is decision making based on belief functions. Here, a system is described that actively collects information by maximizing the expected information gain of possible actions. The belief update as well as the information gain computation are fully formalized using belief function theory and corresponding Monte-Carlo algorithms with linear time and space complexity are presented. In an application to object recognition it is demonstrated that belief functions can be used to make the weakness of statistical evidence resulting from a limited amount of training data explicit. It is shown empirically that an evidential model, which considers the limitation of the data, outperforms a corresponding probabilistic model in terms of recognition rate.

Kurzfassung

Gegenstand dieser Dissertation ist die Belief-Funktionen-Theorie und ihre Anwendung in verschiedenen Kontexten. Die Belief-Funktionen-Theorie (auch als Dempster-Shafer-Theorie bekannt) ist ein mathematisches Framework zur Beschreibung von quantifizierten “beliefs” eines Agenten. Es kann als eine Generalisierung der Bayesianischen Wahrscheinlichkeitstheorie interpretiert werden und ermöglicht es, zwischen verschiedenen Arten von Unsicherheit zu unterscheiden. Insbesondere können Belief-Funktionen Unsicherheit explizit machen, welche aus einem Mangel an Evidenz resultiert. In dieser Arbeit werden Anwendungen der Belief-Funktionen-Theorie sowohl auf einer theoretischen als auch algorithmischen Ebene untersucht. Einer der größten Kritikpunkte in Bezug auf die Verwendung von Belief-Funktionen ist die exponentielle Komplexität ihrer Repräsentation und Kombination. Dieser Kritik wird in der Arbeit begegnet, indem gezeigt wird, wie sich effiziente Algorithmen auf Basis von Monte-Carlo-Approximationen sowie durch Ausnutzung von Unabhängigkeit entwickeln lassen.

Zunächst wird der Kontext von zeitlichen Prozessen betrachtet, die Unsicherheit aufweisen, welche mit Belief-Funktionen modellierbar ist. Hierfür werden Evidenz-basierte zeitliche Aktualisierungsgleichungen hergeleitet, die Bayesianisches Filtern verallgemeinern und die das zeitlich-rekursive Schätzen des Zustandes eines dynamischen Systems ermöglichen. Um die exponentielle Komplexität des Lösen dieser Gleichungen zu verringern, wird eine Monte-Carlo-Approximation vorgestellt, die in einem Evidenz-basierten Partikel-Filter-Algorithmus resultiert. Dieser Evidenz-basierte Partikel-Filter stellt eine Generalisierung von probabilistischen Partikel-Filtern in diskreten Domänen dar und reduziert die exponentielle Zeit- und Platzkomplexität der analytischen Filter-Lösung auf eine Komplexität, welche linear in der Zustandsraumgröße ist.

Der zweite Kontext, welcher in der Arbeit betrachtet wird, ist räumliche Unsicherheit; insbesondere das Problem der simultanen Lokalisierung und Kartierung (simultaneous localization and mapping, SLAM). Bei SLAM hat ein mobiler Roboter, der eine unbekannte Umgebung exploriert, die Aufgabe, eine Karte der Umgebung zu konstruieren und sich gleichzeitig mithilfe dieser Karte zu lokalisieren. In der Arbeit wird bewiesen, dass die Verbundverteilung des Roboter-Pfades und der Karte in eine probabilistische Pfadkomponente und eine Evidenz-basierte Kartenkomponente faktorisiert werden kann. Diese faktorisierte Verbundverteilung wird mittels eines Rao-Blackwellisierten Partikel-Filters

approximiert, was in einem Evidenz-basierten SLAM-Algorithmus resultiert, welcher den populären probabilistischen FastSLAM-Algorithmus generalisiert. Die Grid-Maps, die der Algorithmus erzeugt, werden durch Belief-Funktionen beschrieben und stellen dem Roboter somit zusätzliche Informationen über die Unsicherheit in der Karte zur Verfügung. Der zeitliche Berechnungsaufwand zur Integration einer neuen Messung ist linear in der Anzahl der Grid-Zellen und somit identisch zur Komplexität des probabilistischen FastSLAM-Algorithmus.

Der letzte Kontext, welcher in der Arbeit untersucht wird, ist Entscheidungsfindung basierend auf Belief-Funktionen. In diesem Zusammenhang wird ein System beschrieben, das aktiv Informationen sammelt, indem es den zu erwartenden Informationszuwachs von möglichen Aktionen maximiert. Sowohl die Belief-Aktualisierung als auch die Informationszuwachsberechnung werden vollständig mittels Belief-Funktionen-Theorie formalisiert und entsprechende Monte-Carlo-Algorithmen mit linearer Zeit- und Platzkomplexität werden vorgestellt. In Rahmen einer Anwendung auf Objekterkennung wird gezeigt, dass Belief-Funktionen die Schwäche von statistischer Evidenz, welche aus einem begrenzten Lerndatensatz stammt, explizit machen können. Es wird empirisch gezeigt, dass ein Evidenz-basiertes Modell, welches die Begrenztheit der Daten berücksichtigt, zu höheren Erkennungsraten als ein entsprechendes probabilistisches Modell führt.

Acknowledgments

First of all, I would like to thank my supervisor Kerstin Schill for all the support and for providing me with the opportunity to freely pursue my research interests in this work. To her, I am very grateful for originally inspiring my interest in uncertainty theories and belief function theory in particular. I would also like to thank Günther Palm for kindly agreeing to act as a reviewer for this dissertation and for his insightful remarks about the problem of independence.

I am thankful to all the people I have worked with in the Cognitive Neuroinformatics group and I have always appreciated the friendly work atmosphere. For the interesting discussions related to various aspects of this thesis and for jointly working on publications, I have to thank Joachim Clemens, Tobias Kluth, Niclas Schult, Johannes Wolter, and Christoph Zetsche. To Ingrid Friedrichs, I am very thankful for taking care of all sorts of administrative issues. I am very grateful to Eva-Lisa Meldau, Joachim Clemens, and Tobias Kluth for proof reading several chapters.

This work was in part supported by DFG (SFB/TR8 Spatial Cognition, project A5-[ActionSpace]). Their financial support is gratefully acknowledged.

Finally, I would like to thank my family and particularly Eva for their constant support and encouragement.

Contents

Abstract	3
Kurzfassung	5
Acknowledgments	7
Contents	9
List of Figures	13
Notation	15
1. Introduction	17
1.1. Belief Function Theory	18
1.2. Thesis Contribution	20
1.3. Thesis Outline	22
2. Belief Function Theory	23
2.1. Belief Functions	23
2.1.1. Mass Function	24
2.1.2. Belief Function	25
2.1.3. Plausibility Function	27
2.1.4. Commonality Function	27
2.1.5. Classes of Belief Functions	28
2.2. Combination Rules	29
2.2.1. Dempster's Rule	30
2.2.2. Conjunctive Rule	32
2.2.3. Disjunctive Rule	33
2.2.4. Other Rules	33
2.3. Extension and Marginalization	35
2.4. Conditional Belief Functions	35
2.5. Distinctness and Independence	37
2.6. Generalized Bayesian Theorem	39
2.7. Pignistic Transformation	41
2.8. Uncertainty Measures	41

3. Evidential Particle Filtering	45
3.1. Introduction	45
3.2. Related Work	46
3.2.1. Temporal Updating of Belief Functions	47
3.2.2. Monte-Carlo Approximations of Belief Functions	47
3.3. Evidential Filtering	48
3.3.1. Prediction Step	50
3.3.2. Correction Step	51
3.3.3. Reduction to Bayesian Filtering	52
3.4. Evidential Particle Filtering	53
3.4.1. Prediction Step	54
3.4.2. Correction Step	55
3.4.3. Generating Compatible Samples	58
3.4.4. Derivation	60
3.5. Complexity and Approximation Error	63
3.6. Application to Bearings-only Tracking	66
3.7. Discussion	70
4. Evidential SLAM	73
4.1. Introduction	73
4.2. Simultaneous Localization and Mapping	74
4.2.1. FastSLAM	76
4.3. Evidential FastSLAM	78
4.3.1. Localization	79
4.3.2. Grid Mapping	81
4.3.3. Algorithm	82
4.4. Models	83
4.4.1. Motion Model	84
4.4.2. Forward Sensor Model	85
4.4.3. Inverse Sensor Model	91
4.5. Experimental Results	97
4.5.1. First Experiment	97
4.5.2. Second Experiment	100
4.6. Discussion	102
5. Active Evidential Recognition	107
5.1. Introduction	107
5.2. Related Work	109
5.3. Recognition Architecture	111
5.3.1. Inference	112
5.3.2. Information Gain	113
5.4. Model Learning	117
5.4.1. Maximum Likelihood	120
5.4.2. Laplace Smoothing	120

5.4.3. Imprecise Dirichlet Model	120
5.4.4. Belief Maximization	121
5.4.5. MCD	123
5.4.6. Comparison	125
5.5. Application to Object Recognition	128
5.5.1. Dataset	128
5.5.2. Visual Processing	129
5.5.3. Sensorimotor Model	130
5.5.4. Example	132
5.5.5. Results	132
5.6. Discussion	137
6. Conclusion	139
6.1. Summary	139
6.2. Outlook	141
A. Proofs	145
A.1. Belief-Probability Combination	145
A.2. Belief-Probability Product Rule	146
B. Software	149
B.1. PyDS	149
Own Publications	151
Bibliography	153

List of Figures

2.1. Belief function representations	24
3.1. Dynamic belief network for filtering	49
3.2. Monte-Carlo algorithm for prediction step	54
3.3. Prediction step illustration	55
3.4. Monte-Carlo algorithm for correction step	56
3.5. Correction step illustration	57
3.6. Compatible sample generation algorithm	58
3.7. Sampling illustration	60
3.8. Computation time and approximation error	65
3.9. Marginal position plausibility	69
3.10. Marginal destination plausibility	70
3.11. Tracking error	71
3.12. Mean tracking error	72
4.1. Dynamic belief network for SLAM	75
4.2. Evidential FastSLAM algorithm	83
4.3. Forward sensor model	88
4.4. Importance weight algorithm	91
4.5. Measurement plausibility for localization	92
4.6. Inverse sensor model algorithm	95
4.7. Inverse sensor model	96
4.8. Ground truth for the first experiment	98
4.9. Maps generated in the first experiment	99
4.10. Position error in the first experiment	101
4.11. Ground truth for the second experiment	102
4.12. Maps generated in the second experiment	103
4.13. Position error in the second experiment	104
5.1. Architecture for active evidential recognition	111
5.2. Belief network for classification	112
5.3. Importance sampling combination algorithm	114
5.4. Information gain algorithm	118
5.5. Belief function construction example	125
5.6. Mean classification accuracy on a synthetic dataset	126

5.7. Caltech-256 dataset	129
5.8. Clustered image patches	130
5.9. Sensorimotor feature histograms	131
5.10. Object recognition example	133
5.11. Mean classification accuracy on Caltech-256	134
5.12. Mean classification accuracy for each class	135
5.13. Comparison of information gain with random behavior	136

Notation

Symbol	Reference	Description
$ A $		cardinality of set A
\overline{A}		complement of set A
$A \setminus B$		difference of sets A and B
$\mathcal{P}(A)$		power set of set A
$A_{1:n}$		sequence A_1, A_2, \dots, A_n
$P(A)$		probability of A
$E(X)$		expected value of random variable X
$O(g)$		complexity class g
Θ	2.1	frame of discernment
m	2.1.1	mass function
bel	2.1.2	belief function
pl	2.1.3	plausibility function
q	2.1.4	commonality function
f		general belief function $f \in \{m, bel, pl, q\}$
\oplus	2.2.1	Dempster's rule of combination
Con	2.2.1	weight of conflict
\odot	2.2.2	conjunctive rule of combination
\oslash	2.2.3	disjunctive rule of combination
\otimes	2.4	generic combination rule $\otimes \in \{\oplus, \odot, \dots\}$
η		normalization constant
$f^{\uparrow\Theta_1 \times \Theta_2}$	2.3	vacuous extension to $\Theta_1 \times \Theta_2$
$f^{\downarrow\Theta}$	2.3	marginalization over Θ
$f[A]$	2.4	belief function conditioned on set A
e_0	2.6	prior evidence

(continues on next page)

Symbol	Reference	Description
$BetP$	2.7	pignistic probability function
$H(P)$	2.8	Shannon entropy of probability distribution P
$H_{BetP}(m)$	2.8	pignistic entropy of mass function m
x_t	3.3, 4.2	state at time t
z_t	3.3, 4.2, 5.3	observation/measurement at time t
K	3.4, 4.3.3, 5.3.1	number of samples
$X_t^{[k]}$	3.4	k -th sample at time t
\mathcal{X}_t	3.4	sample set at time t
$\widehat{X}_t^{[k]}$	3.4.1	k -th proposal sample at time t
$\widehat{\mathcal{X}}_t$	3.4	proposal sample set at time t
$\widetilde{X}_t^{[k]}$	3.4.2	k -th compatible sample at time t
$w_t^{[k]}$	3.4.2, 4.3.3	k -th importance weight at time t
u_t	4.2, 5.3	control/action at time t
Y	4.2	map
Y_i	4.2	i -th grid cell
M	4.2	number of grid cells
X	5.3	object class
$I(u_t)$	5.3.2	expected information gain of action u_t
P^-	5.4.3	lower probability function
P^+	5.4.3	upper probability function
$poss$	5.4.5	possibility function

1

Introduction

Uncertainty permeates virtually every aspect of our daily lives, be it the result of sensing ambiguities or of incomplete knowledge. Most of the time, we do not even notice this because we are so adept at handling all these uncertainties. In contrast, when developing intelligent agents that interact with the real world, it becomes apparent that modeling uncertainty is actually a difficult problem. Nonetheless, in order to make these agents act in a robust fashion, they need to be able to cope with the various forms of uncertainty they encounter.

Acknowledging this fact has led to a paradigm shift in fields like robotics where it is now common to employ mathematical formalisms for representing uncertainty [Thrun et al., 2005]. Overall, the role of uncertainty in these fields has changed considerably. Where it was once considered a nuisance that should be avoided as much as possible, it is now embraced as a basis for reliable autonomous behavior. This has to do with increased computational resources but even more so with the availability of appropriate mathematical tools. An agent with an explicit representation of uncertainty knows about the limitations of its knowledge and can therefore better predict the outcome of its actions. In addition, the representation is not a black box to the developer and can be analyzed using statistical methods.

There are a variety of mathematical frameworks for expressing uncertainty [Khaleghi et al., 2013]. Probability theory is the predominant one and is very widely-applied today. Other frameworks include fuzzy set theory [Zadeh, 1965], possibility theory [Zadeh, 1978], rough set theory [Pawlak, 1982], and belief function theory [Shafer, 1976]—the subject of this thesis. The reason why there are multiple frameworks is that there are different types of uncertainty.

In this thesis, the term “uncertainty” generally refers to *epistemic* uncertainty because it corresponds to beliefs held by an agent about the world. When deal-

ing with *aleatory* uncertainty related to randomness and chance, probability theory is usually the preferred framework. Uncertainty resulting from a lack of evidence is of a different nature though. This latter type of uncertainty is the result of ignorance rather than randomness. The Bayesian view is that ignorance can be adequately represented using probability theory by applying the principle of indifference [Keynes, 1921]. This principle states that, in the absence of evidence favoring any particular outcome, the probabilities representing the beliefs for each outcome should be the same for all outcomes. In contrast, belief function theory distinguishes between these types of uncertainty and thus makes ignorance explicit. There are other types of uncertainty (e.g., vagueness of natural language which can be described by fuzzy sets) but these are not considered in this thesis.

1.1. Belief Function Theory

Belief function theory was originally developed by Shafer in his book titled “*A mathematical theory of evidence*” [Shafer, 1976]. It is also known as Dempster-Shafer theory or evidence theory, and the qualifier “evidential” is usually synonymous with “based on belief functions”. Like Bayesian probability theory, it is a theory of quantified beliefs. Central to the theory is the notion of evidence and how different pieces of evidence should be combined in order to make inferences. Belief function theory can be interpreted as a generalization of Bayesian probability theory. Note that, while there are extensions to infinite domains [Smets, 2005a, Dempster, 2001], the belief functions considered here are usually assumed to have finite domains.

Example

Consider the following example: Someone offers you a bet on the outcome of a coin toss. Not knowing the person, there is no reason to trust that the coin is fair. What should your belief about the possible outcomes be in this state of total ignorance? In the Bayesian framework, the principle of indifference dictates that both outcomes are modeled as equiprobable with $P(\text{heads}) = P(\text{tails}) = \frac{1}{2}$. This is also not changed if the parameter of the Bernoulli distribution is itself modeled as a random variable with its own distribution because integration over this parameter nonetheless results in a uniform distribution. The Bayesian belief state of total ignorance is therefore equivalent to a situation where the coin has been tested extensively and is determined to be fair.

In contrast, a belief function can make this state of ignorance explicit by assigning all belief mass to the disjunction of the possible outcomes. It therefore only states that $P(\{\text{heads}, \text{tails}\}) = 1$ and remains entirely agnostic about the true probability distribution. In case the coin is later determined to be fair, the belief can be updated using the new evidence and the belief function

would reduce to a uniform probability distribution. Unlike the Bayesian framework, belief function theory thus allows an agent to distinguish between two fundamentally different states of belief.

History and Interpretations

A detailed account of the historical development of belief function theory is given in [Yager and Liu, 2008]. The theory developed by Shafer builds on previous works by Dempster on lower and upper probabilities [Dempster, 1966, Dempster, 1967, Dempster, 1968]. The term “Dempster-Shafer theory” was coined in [Barnett, 1981], which also introduced the ideas to the broader artificial intelligence community. The work in [Gordon and Shortliffe, 1985] further popularized the theory in the context of expert systems. There have been long debates about the question whether Bayesian probability theory is sufficient for modeling uncertainty or whether belief function theory is more appropriate [Smets, 1992c]. However, many of the arguments are more of a philosophical nature and are not subject of this thesis.

Today, belief function theory encompasses multiple schools of thought, comparisons of which can be found in [Smets, 2000, Kohlas and Monney, 1994]. A rough disjunction can be made between a probabilistic interpretation of belief functions and a non-probabilistic interpretation. Dempster’s original work on one-to-many mappings applied to a probability space leads to lower and upper bounds of probabilities and thus constitutes a probabilistic interpretation. Another example of a probabilistic interpretation is the theory of hints developed by Kohlas [Kohlas and Monney, 2008].

In contrast, Shafer’s approach can be interpreted as being non-probabilistic because it has its own axioms and is not derived from probability theory. However, the existence of some partially-known probability measure corresponding to a belief function is usually still assumed in this case. The transferable belief model (TBM) [Smets and Kennes, 1994, Smets, 1998b, Smets, 1990] developed by Smets goes even further by rejecting the notion of an underlying probability measure altogether. The TBM consists of two levels: a *credal* level where beliefs are represented and combined, and a *pignistic*¹ level where decisions are made based on probabilities derived from the credal level.

The general attitude towards these different interpretations of belief functions is rather pragmatic in this thesis. Overall, it is closest to the TBM framework though because many of the tools used throughout this thesis have been originally developed in the TBM framework (e.g., the generalized Bayesian theorem and the pignistic transformation, see Chap. 2).

¹Derived from the Latin word “pignus” meaning a “bet”.

Computational Complexity

One of the major criticisms directed against belief function theory is its computational complexity. Because belief masses can be assigned to arbitrary subsets of the space under consideration, the complexity of representing and combining belief functions is exponential in the worst case [Orponen, 1990]. To make matters worse, if there are multiple variables, the size of the product space corresponding to these variables is in itself exponential with respect to the number of variables. There are essentially three strategies for reducing the computational complexity associated with belief functions: exploitation of independence, deterministic approximations, and Monte-Carlo approximations. An overview of efficient algorithms for belief functions can be found in [Wilson, 2000].

Like in probability theory, exploiting independence between evidential variables allows inference problems based on multiple variables to be decomposed into smaller subproblems. These can then be solved independently using local computations [Shafer et al., 1987, Shenoy and Shafer, 1990]. With independence, it is thus possible to avoid the exponential complexity resulting from the product space of multiple variables. However, it does not remove the exponential complexity associated with considering all possible subsets of a space.

Deterministic approximations of belief functions are usually based on restricting the number of subsets that are allowed to receive belief mass. Examples include hierarchical hypothesis spaces [Gordon and Shortliffe, 1985, Schill, 1997] and lattice structures [Dencœux and Masson, 2012]. While resulting in efficient computations, the disadvantage is that only certain classes of belief functions can be expressed under such restrictions.

In contrast, Monte-Carlo approaches use a finite number of samples to approximate a belief function [Moral and Salmerón, 1999, Moral and Wilson, 1996]. The advantage over deterministic approximations is that sampling-based approximations are non-parametric and can therefore represent a much larger class of belief functions. The extent of the resulting approximation error depends on the number of samples and on how “spread-out” the true belief distribution is. For example, if the belief masses are concentrated on a small number of subsets, the approximation error tends to be negligible.

1.2. Thesis Contribution

This thesis explores applications of belief function theory to three fundamental domains: time, space, and action. Any embodied agent must be proficient in all of these domains, i.e., it must be able reason temporally as well as spatially, and it must be able to make decisions. In particular, it must be able to do these things while faced with different types of uncertainty. This thesis explores each of the three domains by demonstrating how belief function theory can be applied in order to model the underlying uncertainties. Within each domain,

the contributions of this thesis are both theoretical and algorithmic.

Time: State estimation over time is an essential problem because environments are generally not static. In this case, uncertainty arises from both limited observability of states and from the unpredictability of state changes. Probabilistic methods like Kalman filters and particle filters are well-established methods for state estimation. In this thesis, probabilistic particle filtering is generalized in the belief function framework, allowing ignorance in the underlying models to be represented explicitly. On the theoretical side, this requires deriving equations for evidential filtering that generalize Bayesian filtering for discrete states. Due to the intractability of these equations for larger state spaces, an efficient Monte-Carlo algorithm is presented which reduces the computational complexity from exponential to linear in the state space size.

Space: Obtaining a spatial representation of an environment from uncertain measurements is an essential capability of autonomous mobile agents. In the robotics community, this problem is referred to as *simultaneous localization and mapping* (SLAM) because a robot exploring its environment needs to localize itself while creating a map at the same time. Probabilistic SLAM algorithms, usually based on Kalman or particle filters, have become quite mature over the last decade and are now a standard tool for mobile robots. In this thesis, an evidential SLAM algorithm is proposed which generates evidential grid maps. These evidential grid maps contain additional dimensions of uncertainty compared to probabilistic grid maps and therefore provide the mobile robot with additional information about the environment. The theoretical contribution in this case is a factorization of the joint distribution over the robot's path and the map. Based on this factorization, a Rao-Blackwellized particle filter algorithm is developed which allows the robot to efficiently approximate the joint distribution. In addition, efficient evidential forward and inverse sensor models for sonar are presented.

Action: An agent interacting with its environment needs to be able to make rational decisions under uncertainty. Typically, this is done in a Bayesian framework by maximizing the expected value of a utility function. In this thesis, an architecture is presented which actively gathers evidence by performing actions that maximize the expected information gain with respect to the current belief state. This principle is demonstrated in an application to object recognition where the unreliability of parameter estimates, caused by a small training set, is modeled by belief functions. The contribution in this case is a formalization of the inference process and of the information gain maximization within the TBM framework. Following the two-level approach of the TBM framework, inference is conducted at the credal level and actions are selected at the pignistic level by maximizing the expected information gain. Both for inference and for information gain computation, efficient Monte-Carlo algorithms with linear time and

space complexity are presented.

The algorithms in this thesis are generally based on exploiting independence properties and on Monte-Carlo approximations. As a result, the algorithms tend to scale well to large spaces (e.g., the evidential SLAM approach can easily handle maps consisting of 50,000+ variables). All of the core algorithms presented in this thesis are publicly available in an open-source library. Aside from theoretical and algorithmic contributions, the thesis also provides empirical comparisons of the proposed approaches to corresponding Bayesian approaches. Finally, the thesis contains a comprehensive compilation of the theoretical basics of belief function theory. This in itself could be useful because one of the problems hindering a broader adoption of belief function theory is a lack of comprehensive text books.

1.3. Thesis Outline

The thesis is structured into four main chapters. In Chap. 2, the belief function theory formalism that underlies all other chapters is introduced. The evidential particle filter is described in Chap. 3, the evidential SLAM approach in Chap. 4, and the active recognition architecture in Chap. 5. A summary and outlook are provided in Chap. 6. Finally, Appendix A contains mathematical proofs and Appendix B describes the open source software developed in the context of this thesis.

2

Belief Function Theory

This chapter introduces the mathematical formalism underlying belief function theory. Basic definitions, different representations of belief functions, and rules for combining belief functions are presented, among other things. The chapter forms the theoretical basis for all subsequent chapters and is therefore frequently referred to.

2.1. Belief Functions

A *frame of discernment* Θ is a finite set of mutually exclusive elements in a domain. A hypothesis or proposition is a subset $A \subseteq \Theta$ of the frame of discernment, i.e., it is an element of the power set $\mathcal{P}(\Theta)$. A hypothesis consisting of only one element ($A \subseteq \Theta$ with $|A| = 1$) is called a *singleton*.

A *belief function* bel assigns a belief value to each hypothesis based on one or more pieces of evidence. In contrast to the Bayesian probability framework, in belief function theory, additivity of belief values is not required. This means that the belief in a hypothesis and the belief in its complement can be less than 1.

$$bel(A) + bel(\bar{A}) \leq 1, \quad \forall A \subseteq \Theta \quad (2.1)$$

This is a result of the fact that belief mass can be freely assigned to any hypothesis $A \subseteq \Theta$ without committing mass to the subsets $B \subset A$. Because of this freedom, belief functions provide an additional “dimension of uncertainty”. Instead of just having singletons with different probabilities like in the Bayesian framework, the cardinality of each hypothesis receiving a direct belief assignment can vary. This additional dimension of uncertainty allows belief functions to make *ignorance* explicit. For example, by assigning a belief value to the set

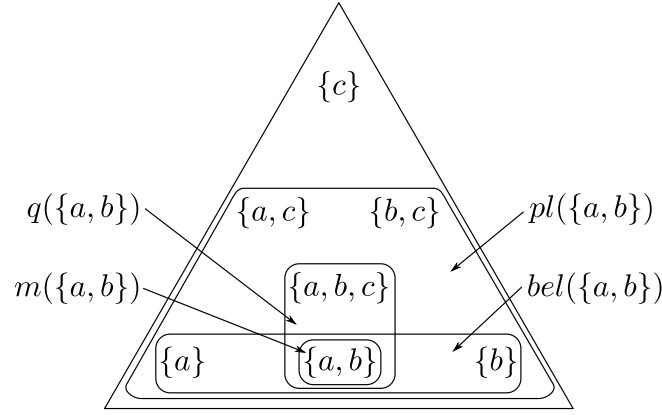


Figure 2.1.: Illustration of different belief function representations. The frame of discernment Θ consists of three elements a, b, c in this example and the triangle contains all subsets of Θ except for \emptyset . The indicated areas correspond to the mass of the different belief representations m , bel , pl , and q associated with the set $\{a, b\}$.

$A = \{a, b\}$, one abstains from making any statement about whether a is more “probable” than b . Instead, such an assignment expresses complete ignorance about this question.

There are several equivalent representations for quantifying belief within the belief function framework. The four most important representations are mass functions denoted by m , belief functions denoted by bel , plausibility functions denoted by pl , and commonality functions denoted by q . An illustration of these representations is shown in Fig. 2.1. All these representations convey exactly the same information because they are in one-to-one correspondence to each other. The term “belief function” is somewhat ambiguous because it is used both as a general term (encompassing all the different representations) and as a specific term referring to the bel representation. In this work, the term is used in the general sense unless explicitly stated otherwise. The following sections introduce the different belief representations in a formal manner. In addition, rules for converting between the representations are defined as well as the most important classes of belief functions.

2.1.1. Mass Function

A *mass function* (also called *basic belief assignment* or *basic probability assignment*) is in many respects the most fundamental belief representation and all other representations can be easily obtained from a mass function. Formally, a mass function m is a mapping $m : \mathcal{P}(\Theta) \rightarrow [0, 1]$ assigning a mass value to each

hypothesis $A \subseteq \Theta$ of the frame of discernment Θ such that

$$\sum_{A \subseteq \Theta} m(A) = 1. \quad (2.2)$$

The value $m(A)$ is the amount of belief strictly committed to hypothesis A . Such an assignment to a set A implies ignorance about the belief distribution over subsets of A .

In Shafer's original work [Shafer, 1976], there is an additional constraint requiring that a mass function must not assign a positive value to the empty set.

$$m(\emptyset) = 0 \quad (2.3)$$

A mass function satisfying this property is called *normalized*. This constraint is absent in the TBM framework where the mass assigned to \emptyset usually represents the possibility that the true value is not included in the frame of discernment. Smets therefore argues that requiring $m(\emptyset) = 0$ corresponds to a closed-world assumption while allowing $m(\emptyset) > 0$ corresponds to an open-world assumption [Smets, 1992b, Smets, 1988].

All models presented in this work are based on a closed-world assumption. This means the frame of discernment is assumed to be exhaustive. As a result, belief functions are usually normalized. There are some exceptions where unnormalized belief functions are used (see Chap. 4). However, in this case, the use of unnormalized belief functions mainly serves as a way of explicitly representing conflict between different pieces of evidence. Regardless of the purpose, it is always possible to turn an unnormalized mass function m into a normalized mass function m' .

$$m'(A) = \begin{cases} \frac{m(A)}{1-m(\emptyset)} & \text{if } A \neq \emptyset \\ 0 & \text{if } A = \emptyset \end{cases} \quad (2.4)$$

Here, the factor $(1 - m(\emptyset))^{-1}$ acts as a normalization constant.

Often, one is only interested in the sets $A \subseteq \Theta$ with positive mass values $m(A) > 0$. Such sets are called *focal sets*. The set \mathcal{F}_m consisting of all focal sets corresponding to a mass function m is defined as

$$\mathcal{F}_m = \{A | A \subseteq \Theta, m(A) > 0\}. \quad (2.5)$$

(Remark: Sets are generally denoted by capital letters while their elements are denoted by lower-case letters. For singletons, the bracket notation $m(\{a\})$ can become quite cumbersome which is why set brackets are usually omitted for singletons: $m(a) = m(\{a\})$ with $a \in \Theta$.)

2.1.2. Belief Function

The total amount of belief committed to a hypothesis $A \subseteq \Theta$, including all subsets $B \subseteq A$, is denoted by $bel(A)$. The function $bel : \mathcal{P}(\Theta) \rightarrow [0, 1]$ is called

a *belief function* (understood in the specific rather than in the general sense of the term). It can be directly computed from a mass function m .

$$bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B), \quad \forall A \subseteq \Theta, A \neq \emptyset \quad (2.6)$$

$$bel(\emptyset) = 0 \quad (2.7)$$

If m is normalized, a direct consequence of Eq. (2.6) is

$$bel(\Theta) = 1. \quad (2.8)$$

The property of subadditivity of a belief function bel was already defined in Eq. (2.1). The inequality $bel(A) + bel(\overline{A}) < 1$ holds whenever there is a focal set B that is neither a subset of A nor of \overline{A} (for example, if $m(\Theta) > 0$ and $A \neq \Theta$).

A belief function bel is sometimes interpreted as defining a “lower bound” for an unknown probability function P , although this interpretation is only valid under specific circumstances (see Sect. 1.1). However, for a given belief function bel , one can always find a probability function P such that

$$bel(A) \leq P(A), \quad \forall A \subseteq \Theta. \quad (2.9)$$

In this case, bel and P are called *compatible*.

When dealing with unnormalized mass functions, it is sometimes more convenient to use the so-called *implicability function* b instead of bel . The only difference is that it includes the mass assigned to the empty set.

$$b(A) = bel(A) + m(\emptyset) = \sum_{B \subseteq A} m(B), \quad \forall A \subseteq \Theta \quad (2.10)$$

Its interpretation is less intuitive though and it mostly serves a notational purpose.

As stated above, all the different belief representations are in one-to-one correspondence. Just as it is possible to obtain a belief function bel from a mass function m using Eq. (2.6), it is also possible to recover a mass function from a belief function bel (or b if the beliefs are not normalized). This conversion is based on the Fast Möbius Transformation [Thoma, 1989].

$$m(A) = \sum_{B \subseteq A} (-1)^{|A \setminus B|} b(B), \quad \forall A \subseteq \Theta \quad (2.11)$$

Such conversions are particularly useful if a certain representation simplifies the computations associated with an operation like evidence combination (see Sect. 2.2).

2.1.3. Plausibility Function

The *plausibility* $pl(A)$ is the amount of belief *not* strictly committed to the complement \bar{A} . It therefore expresses how *plausible* a hypothesis A is, i.e., how much belief mass potentially supports A . On a formal level, a *plausibility function* $pl : \mathcal{P}(\Theta) \rightarrow [0, 1]$ is defined as

$$pl(A) = bel(\Theta) - bel(\bar{A}), \quad \forall A \subseteq \Theta. \quad (2.12)$$

For the special case of normalized plausibility functions where $bel(\Theta) = 1$, this definition reduces to

$$pl(A) = 1 - bel(\bar{A}), \quad \forall A \subseteq \Theta. \quad (2.13)$$

The plausibility $pl(A)$ can be computed from a mass function m in the following way:

$$pl(A) = \sum_{B \subseteq \Theta, B \cap A \neq \emptyset} m(B), \quad \forall A \subseteq \Theta, \quad (2.14)$$

$$pl(\emptyset) = 0. \quad (2.15)$$

Whereas bel can be viewed as a lower bound for an unknown probability function P under a lower- and upper probability interpretation, the plausibility can be viewed as an upper bound.¹ For a normalized plausibility function pl , a compatible probability function P must satisfy the property

$$pl(A) \geq P(A), \quad \forall A \subseteq \Theta. \quad (2.16)$$

2.1.4. Commonality Function

The *commonality* $q(A)$ states how much mass in total is committed to A and all of the supersets B with $A \subseteq B \subseteq \Theta$. The commonality $q(A)$ therefore expresses how much mass potentially supports the *entire* set A . A *commonality function* $q : \mathcal{P}(\Theta) \rightarrow [0, 1]$ is defined as

$$q(A) = \sum_{B \subseteq \Theta, B \supseteq A} m(B), \quad \forall A \subseteq \Theta. \quad (2.17)$$

In order to compute a mass function m from a given commonality function q , the following equation can be used:

$$m(A) = \sum_{B \subseteq \Theta, B \supseteq A} (-1)^{|B \setminus A|} q(B), \quad \forall A \subseteq \Theta. \quad (2.18)$$

¹In fact, Shafer refers to pl as the “upper probability function” [Shafer, 1976, chapter 2].

2.1.5. Classes of Belief Functions

Regardless of the representation, there are certain classes of belief functions that are particularly important. These classes are defined in the following.

Categorical Belief Function

A belief function is called *categorical* if it is normalized and has only one focal set.

$$m(A) = 1, \quad A \subseteq \Theta \quad (2.19)$$

Vacuous Belief Function

A belief function is called *vacuous* if it is categorical and the focal set is the frame of discernment Θ .

$$m(\Theta) = 1 \quad (2.20)$$

A vacuous belief function represents a state of total ignorance.

Simple Belief Function

A *simple* belief function has at most two focal sets: the frame of discernment Θ and a strict subset of Θ .

$$m(A) = 1 - w, \quad A \subset \Theta, 0 \leq w \leq 1 \quad (2.21)$$

$$m(\Theta) = w \quad (2.22)$$

Dogmatic Belief Functions

A belief function is called *dogmatic* if the frame of discernment Θ is not a focal set.

$$m(\Theta) = 0 \quad (2.23)$$

Consonant Belief Function

A belief function is called *consonant* if its focal sets are nested. This means there exists an ordering of all the focal sets, such that

$$A_0 \subset A_1 \subset \dots \subset A_k, \quad \text{with } A_i \in \mathcal{F}_m. \quad (2.24)$$

As a result, a consonant belief function only has up to $|\Theta| + 1$ focal sets.

The corresponding belief and plausibility functions satisfy the following properties [Shafer, 1976, chapter 10]:

$$bel(A \cap B) = \min(bel(A), bel(B)), \quad \forall A, B \subseteq \Theta, \quad (2.25)$$

$$pl(A \cup B) = \max(pl(A), pl(B)), \quad \forall A, B \subseteq \Theta. \quad (2.26)$$

Note that the plausibility function defines a possibility measure in this case (see Sect. 5.4.5).

Bayesian Belief Function

A belief function is called *Bayesian* if it is normalized and all of its focal sets are singletons, i.e., if it is a probability function. In this case, the belief function bel satisfies the property of additivity defined by the Kolmogorov axioms [Kolmogorov, 1933].

$$bel(A \cup B) = bel(A) + bel(B) \text{ if } A \cap B = \emptyset \text{ with } A, B \subseteq \Theta \quad (2.27)$$

In addition, the following equalities hold for Bayesian belief functions:

$$P(A) = bel(A) = pl(A), \quad \forall A \subseteq \Theta, \quad (2.28)$$

$$P(a) = m(a) = q(a), \quad \forall a \in \Theta. \quad (2.29)$$

2.2. Combination Rules

In order to solve inference problems, belief functions representing different pieces of evidence need to be combined in a meaningful way. This is why combination rules are a major building block of belief function theory. Typically, each piece of evidence is represented by a separate belief function. Combination rules are then used to successively fuse all these belief functions in order to obtain a belief function representing all available evidence.

There exist many different types of combination rules within the belief function framework. Surveys of different combination rules are given in [Smets, 2007] and [Sentz and Ferson, 2002]. The most important one is arguably Dempster's rule. Most other combination rules are variations of Dempster's rule and only differ in how they handle conflicting evidence. Generalizations of combination rules that can be parametrized are described in [Smets, 1997] and [Lefevre et al., 2002].

One of the reasons why new combination rules kept getting proposed over time was Zadeh's criticism of Dempster's rule when faced with highly conflicting evidence. In Zadeh's example [Zadeh, 1979, Zadeh, 1984], two doctors independently diagnose the same patient, resulting in two (Bayesian) belief functions m_1 and m_2 where the frame of discernment $\{a, b, c\}$ consists of three possible diseases.

$$\begin{aligned} m_1(a) &= 0.99 & m_1(b) &= 0.01 \\ m_2(c) &= 0.99 & m_2(b) &= 0.01 \end{aligned}$$

The result of combining these two belief functions using Dempster's rule (see Sect. 2.2.1) is $m(b) = 1$, meaning that there is absolute certainty that the patient suffers from disease b . This was interpreted as being counterintuitive by some people because both doctors believe that b is highly unlikely. Combination rules handling this conflict in a different way are the conjunctive rule proposed by

Smets, Yager’s rule, and the rule proposed by Dubois and Prade. It should be noted that Zadeh’s criticism is not really limited to belief functions but can be used just as well against probability theory.

The following sections introduce all combination rules used in this thesis. Dempster’s rule is the most important one because it is used throughout this work. The conjunctive and disjunctive rules are central elements in the TBM framework with the conjunctive rule being identical to Dempster’s rule apart from normalization. All other rules (Yager’s rule, Dubois and Prade’s rule, and the cautious rule) are only used in Chap. 4 where they are compared to Dempster’s rule and the conjunctive rule for a particular application.

2.2.1. Dempster’s Rule

Dempster’s rule of combination was first introduced in [Dempster, 1967] and then reinterpreted by Shafer as a basis for belief function theory. It allows combining normalized belief functions that are defined over the same frame of discernment and are induced by “distinct bodies of evidence” (see Sect. 2.5 for a discussion of the notion of “distinctness”). The resulting belief function reflects a conjunctive combination of the underlying evidence.

Let m_1 and m_2 be normalized mass functions induced by distinct pieces of evidence which are defined over the same frame of discernment Θ . The mass function $m_{1\oplus 2} = m_1 \oplus m_2$ combined according to Dempster’s rule \oplus is defined as

$$m_{1\oplus 2}(A) = \eta \sum_{B \cap C = A} m_1(B) m_2(C), \quad \forall A \subseteq \Theta, A \neq \emptyset, \quad (2.30)$$

$$m_{1\oplus 2}(\emptyset) = 0, \quad (2.31)$$

$$\eta^{-1} = 1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C). \quad (2.32)$$

Here, η is a normalization constant assuring that the resulting mass function is normalized. It accounts for the products of mass values corresponding to all empty intersections of focal sets. Dempster’s rule is commutative, associative, and possesses a neutral element with the vacuous belief function. In contrast to most other rules, there exist a variety of theoretical justifications for the appropriateness of Dempster’s rule for combining evidence [Pichon and Dencœux, 2010, Dubois and Prade, 1986a, Shafer and Tversky, 1985].² In case there are more than two pieces of evidence, the corresponding belief functions are simply successively combined where the order of combination is irrelevant.

When using normalized belief functions, a situation can occur in which two belief functions entirely contradict each other. This happens if all pairwise intersections of focal sets from the two mass functions are empty. In this case,

²These justifications extend to the conjunctive rule which is almost identical.

the belief functions are said to be *flatly contradictory* and their combination is undefined.³

The normalization constant η turns out to be more than just a nuisance because it represents the amount of conflict between two belief functions. The larger η becomes, the higher the conflict there is between m_1 and m_2 . Shafer defines the *weight of conflict* Con associated with the combination of two belief functions as the logarithm $\log(\eta)$ [Shafer, 1976, chapter 3].

$$Con(m_1, m_2) = -\log\left(1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)\right) \quad (2.33)$$

The generalization to more than two belief functions is straightforward. The weight of conflict Con is 0 if there are no empty intersections of focal sets and it is infinite for flatly contradictory belief functions. In particular, the weight of conflict is additive.

$$Con(m_1, \dots, m_{n+1}) = Con(m_1, \dots, m_n) + Con(m_1 \oplus \dots \oplus m_n, m_{n+1}) \quad (2.34)$$

Dempster's rule is often interpreted as a generalization of Bayes' rule. The reason for this interpretation becomes apparent when considering the combination of two Bayesian belief functions. Let e_1, e_2 denote two pieces of evidence. By applying Bayes' rule twice and assuming conditional independence between e_1, e_2 given x , one has:

$$P(x|e_1, e_2) \propto P(e_1|x) P(e_2|x) P(x) \propto \frac{P(x|e_1)}{P(x)} \frac{P(x|e_2)}{P(x)} P(x). \quad (2.35)$$

Ignoring the prior $P(x)$ at the very end (the prior constitutes a separate piece of evidence in the belief function framework), this product corresponds to a special case of Dempster's rule. By setting $m_i(x) \propto P(x|e_i)/P(x)$, each mass function m_i represents the (prior-free) belief induced by evidence e_i . In this case, the combination $m_1 \oplus m_2$ is equal to the posterior probability distribution $P(x|e_1, e_2)$.

Although in most cases Dempster's rule is applied to mass functions, commonality functions are actually a more convenient form of representation for Dempster's rule. Let q_1, q_2 be the commonality functions corresponding to the mass functions m_1, m_2 and let $q_{1 \oplus 2}$ be the commonality function corresponding to the mass function $m_1 \oplus m_2$. Dempster's rule can then be expressed in the following way:

$$q_{1 \oplus 2}(A) = \eta q_1(A) q_2(A), \quad \forall A \subseteq \Theta, A \neq \emptyset \quad (2.36)$$

$$q_{1 \oplus 2}(\emptyset) = 1 \quad (2.37)$$

$$\eta^{-1} = \sum_{B \subseteq \Theta, B \neq \emptyset} (-1)^{|B|+1} q_1(B) q_2(B) \quad (2.38)$$

³The same can happen with probability functions when applying the Bayesian theorem if all prior-likelihood products are 0.

It turns out that for commonality functions, Dempster's rule of combination reduces to computing a simple product of commonality values where η is a normalization constant equal to the one defined in Eq. (2.32). Such a product can be computed more efficiently than the expression in Eq. (2.30). The question in this case is whether a potential conversion to commonalities is worth the reduced computational effort of the combination.

2.2.2. Conjunctive Rule

The conjunctive rule of combination is an adaptation of Dempster's rule and plays a central role in the TBM framework [Smets and Kennes, 1994]. Because the TBM framework explicitly allows unnormalized belief functions, the normalization step performed by Dempster's rule is omitted. Otherwise it is identical to Dempster's rule.

Let m_1 and m_2 be two (possibly unnormalized) mass functions induced by distinct pieces of evidence and which are defined over the same frame of discernment Θ . The mass function $m_{1\odot 2} = m_1 \odot m_2$ resulting from the combination using the conjunctive rule \odot is defined as

$$m_{1\odot 2}(A) = \sum_{B \cap C = A} m_1(B) m_2(C), \quad \forall A \subseteq \Theta. \quad (2.39)$$

Other than for Dempster's rule, the result of the conjunctive rule of combination is always defined. For flatly contradictory evidence, the result is simply $m_{1\odot 2}(\emptyset) = 1$.

Just like Dempster's rule can be efficiently computed in terms of commonality functions, the same applies to the conjunctive rule. Let q_1, q_2 , and $q_{1\odot 2}$ be the commonality functions corresponding to the mass functions m_1, m_2 , and $m_{1\odot 2}$ respectively. Then the conjunctive combination of q_1 and q_2 is defined as

$$q_{1\odot 2}(A) = q_1(A) q_2(A), \quad \forall A \subseteq \Theta. \quad (2.40)$$

As discussed in Sect. 2.1.1, the question whether normalization should be performed usually depends on whether one makes an open- or a closed-world assumption. Under an open-world assumption, normalization should not be performed and the conjunctive rule of combination should be used instead of Dempster's rule. Under a closed-world assumption, the need for normalization depends on the application. On the one hand, a positive mass value $m(\emptyset)$ provides useful information even for a closed-world assumption. This is because the conflict between the underlying pieces of evidence is described by the mass on \emptyset with $Con(m_1, m_2) = -\log(1 - m_{1\odot 2}(\emptyset))$. In contrast to Dempster's rule, this conflict information is preserved when performing multiple combinations even if the original belief functions are not available anymore. If such information is useful for an application (e.g., see Chap. 4), normalization should be avoided.

On the other hand, if one is not interested in the amount of conflict, normalization should be performed and Dempster's rule is more appropriate. In

particular, normalization should be performed if there is significant conflict and belief functions are combined recursively over time. An example of such a situation is the particle filter presented in Chap. 3, where omitting normalization would cause $m(\emptyset)$ to quickly dominate all other mass values, meaning almost all particles would represent \emptyset .

2.2.3. Disjunctive Rule

The disjunctive rule of combination [Dubois and Prade, 1986b] is applied when only one of several pieces of evidence holds. Whereas Dempster's rule and the conjunctive rule correspond to an "and"-like operation, the disjunctive combination rule represents an "or"-like operation. However, the main use of the disjunctive rule is in the context of conditional belief functions, see Sect. 2.4.

Let m_1 and m_2 be two (possibly unnormalized) mass functions induced by distinct pieces of evidence which are defined over the same frame of discernment Θ . The mass function $m_{1\oplus 2} = m_1 \oplus m_2$ resulting from the combination using the disjunctive rule \oplus is defined as

$$(m_1 \oplus m_2)(A) = \sum_{B \cup C = A} m_1(B) m_2(C), \quad \forall A \subseteq \Theta. \quad (2.41)$$

Because the union $B \cup C$ is never empty unless both focal sets are empty, there is no conflict resulting from the disjunctive rule of combination and therefore no need for normalization.

Just like commonality functions can be used for efficiently computing Dempster's rule/the conjunctive rule, implicability functions can be used to express the disjunctive rule in terms of a simple product. Let b_1, b_2 be the implicability functions corresponding to the mass functions m_1, m_2 and let $b_{1\oplus 2}$ be the implicability function corresponding to the mass function $m_1 \oplus m_2$. The disjunctive rule of combination is then defined as the product

$$b_{1\oplus 2}(A) = b_1(A) b_2(A), \quad \forall A \subseteq \Theta. \quad (2.42)$$

2.2.4. Other Rules

Yager's Rule

In [Yager, 1987], Yager proposes a combination rule that assigns the mass associated with conflicting focal sets to the frame of discernment (instead of performing normalization or assigning it to \emptyset). Let m_{12} denote the result of combining two mass functions m_1, m_2 induced by distinct pieces of evidence using Yager's rule. Expressed in terms of the conjunctive rule, Yager's rule is defined as

$$m_{12}(A) = \begin{cases} m_{1\otimes 2}(A) & \forall A \subset \Theta, A \neq \emptyset, \\ m_{1\otimes 2}(\Theta) + m_{1\otimes 2}(\emptyset) & \text{if } A = \Theta, \\ 0 & \text{if } A = \emptyset. \end{cases} \quad (2.43)$$

This means that highly conflicting evidence leads to a state of high ignorance with all conflict-related mass $m_{1\odot 2}(\emptyset)$ being assigned to Θ .

Dubois and Prade's Rule

The rule proposed by Dubois and Prade in [Dubois and Prade, 1986b] is similar to Yager's rule. However, instead of assigning the mass of conflicting focal sets to the frame of discernment, it is assigned to the union of the corresponding focal sets. Let m_{12} denote the result of combining two mass functions m_1, m_2 induced by distinct pieces of evidence using Dubois and Prade's rule. Expressed using the conjunctive rule of combination, the rule is defined as

$$m_{12}(A) = \begin{cases} m_{1\odot 2}(A) + \sum_{B \cap C = \emptyset, B \cup C = A} m_1(B) m_2(C) & \forall A \subseteq \Theta, A \neq \emptyset, \\ 0 & \text{if } A = \emptyset. \end{cases} \quad (2.44)$$

Note that for $|\Theta| = 2$, Yager's rule and Dubois and Prade's rule are identical.

Cautious Rule

The *cautious* combination rule was introduced in [Denœux, 2008] and differs significantly from all the other rules presented here. Whereas the other rules all assume that the underlying pieces of evidence are “distinct”, the cautious rule allows combining non-distinct/overlapping pieces of evidence. The rule shown below is also referred to as the “cautious conjunctive rule” because there also exists a disjunctive version that is not considered here.

Before defining the actual combination rule, some additional concepts and notation need to be introduced. Let A^w denote the simple belief function defined by $m(A) = 1 - w$ and $m(\Theta) = w$. In [Smets, 1995], it is shown that any non-dogmatic belief function can be uniquely represented as a conjunctive combination of simple belief functions $A^{w(A)}$ with

$$m = \odot_{A \subseteq \Theta} A^{w(A)}, \quad (2.45)$$

$$w(A) = \prod_{B \supseteq A} q(B)^{-1^{|B| - |A| + 1}}, \quad \forall A \subseteq \Theta, \quad (2.46)$$

where $w(A) : \mathcal{P}(\Theta) \setminus \Theta \rightarrow [0, +\infty]$ is a weight function that can be computed using the commonality function q belonging to mass function m .

Let m_1, m_2 be two non-dogmatic mass functions (possibly induced by non-distinct pieces of evidence). Let w_1, w_2 be the corresponding weight functions and let m_{12} denote the result of combining m_1 with m_2 using the cautious rule.

$$m_{12} = \odot_{A \subseteq \Theta} A^{\min(w_1(A), w_2(A))} \quad (2.47)$$

The cautious rule is thus computed in terms of a combined weight function $A \mapsto \min(w_1(A), w_2(A))$ where the min-operator corresponds to a conjunctive combination of w_1 and w_2 .

2.3. Extension and Marginalization

When combining two belief functions that are defined over different frames of discernment, both first need to be extended to the joint space. Let Θ_1, Θ_2 be the non-overlapping frames of discernment of mass functions m_1, m_2 respectively. The combination of m_1 and m_2 using, for example, Dempster's rule is then defined over the joint frame of discernment $\Theta_1 \times \Theta_2$. The respective *extensions* of m_1 and m_2 required prior to this combination are defined as

$$m_1^{\uparrow\Theta_1 \times \Theta_2}(A \times \Theta_2) = m_1(A), \quad \forall A \subseteq \Theta_1, \quad (2.48)$$

$$m_2^{\uparrow\Theta_1 \times \Theta_2}(\Theta_1 \times B) = m_2(B), \quad \forall B \subseteq \Theta_2. \quad (2.49)$$

In order to keep notation simple, extensions are omitted throughout the text whenever possible and combined mass functions are implicitly assumed to be defined over their respective joint space. This also means that whenever set operations are performed over a joint space, the operands are implicitly assumed to be extended first (e.g., $A \cap B$ with $A \subseteq \Theta_1, B \subseteq \Theta_2$ actually means $(A \times \Theta_2) \cap (\Theta_1 \times B)$). Furthermore, the notation A, B is used to denote $A \cap B$.

Conversely to extension, a belief function defined over some joint frame of discernment can be *marginalized*. Let mass function m_{12} be defined over $\Theta_1 \times \Theta_2$. The marginal mass functions $m_{12}^{\downarrow\Theta_1}$ and $m_{12}^{\downarrow\Theta_2}$ defined over Θ_1 and Θ_2 respectively are defined as

$$m_{12}^{\downarrow\Theta_1}(A) = \sum_{B \subseteq \Theta_2} m_{12}(A \times B), \quad \forall A \subseteq \Theta_1, \quad (2.50)$$

$$m_{12}^{\downarrow\Theta_2}(B) = \sum_{A \subseteq \Theta_1} m_{12}(A \times B), \quad \forall B \subseteq \Theta_2. \quad (2.51)$$

Just like extension is usually omitted for notational simplicity, marginalization is omitted as well.

The extension defined in Eq. (2.48) and (2.49) is also referred to as a *vacuous extension* because marginalizing on the “newly added space” always yields a vacuous belief function:

$$m_1^{\uparrow\Theta_1 \times \Theta_2 \downarrow \Theta_2}(\Theta_2) = 1, \quad (2.52)$$

$$m_2^{\uparrow\Theta_1 \times \Theta_2 \downarrow \Theta_1}(\Theta_1) = 1. \quad (2.53)$$

2.4. Conditional Belief Functions

Like probability functions, belief functions can be conditioned. If a subset $A \subset \Theta$ of the frame of discernment is known to be true with absolute certainty, then this fact can be expressed using the categorical mass function $m_A(A) = 1$. Let m denote the mass function describing the belief prior to learning that subset

A is true. This prior belief can be conditioned with A by combining m with m_A using Dempster's rule⁴, which results in the conditional mass function $m[A]$:

$$m[A] = m \oplus m_A \text{ with } m_A(A) = 1. \quad (2.54)$$

(Remark: Writing $m[A]$ opposed to $m(\cdot|A)$ has become the standard notation for conditional belief functions because operations often require the entire function as input. For example, it simplifies the notation for combinations where one can write $m[A] \oplus m[B]$ instead of $m(\cdot|A) \oplus m(\cdot|B)$.)

Plugging in the definition of Dempster's rule given by Eq. (2.30) yields the following expression for $m[A]$:

$$m[A](B) = \begin{cases} \eta \sum_{C \subseteq \bar{A}} m(B \cup C) & \forall B \subseteq A, B \neq \emptyset, \\ 0 & \text{otherwise,} \end{cases} \quad (2.55)$$

$$\eta = pl(A)^{-1}. \quad (2.56)$$

Eq. (2.55) is referred to as *Dempster's rule of conditioning*. The corresponding plausibility function $pl[A]$ is given by

$$pl[A](B) = \frac{pl(A \cap B)}{pl(A)}, \quad \forall B \subseteq \Theta. \quad (2.57)$$

Usually, when dealing with conditional belief functions $m[A](B)$, A and B belong to different frames of discernment. In this case, extension and marginalization are implicitly assumed such that $m[A](B)$ is a belief function defined over the frame of discernment corresponding to B .

Conditional belief functions can also be used to make evidence explicit in the notation. Given some piece of evidence e , the induced mass function can be written as $m[e]$. As a result, a combination of two pieces of evidence e_1, e_2 based on a rule \ast with $\ast \in \{\oplus, \odot, \ominus, \dots\}$ can be written with the evidence made explicit:

$$m[e_1, e_2] = m[e_1] \ast m[e_2]. \quad (2.58)$$

Assuming one has a set of belief functions with frame of discernment Θ , each conditioned by an element from a different frame of discernment Ω , the disjunctive rule of combination allows one to construct a belief function conditioned by an arbitrary subset of Ω . This is very useful in practice because one oftentimes is only provided with singleton-conditioned belief functions when, instead, one needs a belief function conditioned by a larger set. Let $\{f[a_i] | 1 \leq i \leq k\}$ be a set of conditional belief functions defined over Θ with $f \in \{m, bel, pl, q\}$ and $a_i \in \Omega$. If the evidence underlying each belief function $f[a_i]$ is distinct, the disjunctive rule of combination can then be used to construct the belief function $f[A]$ with $A \subseteq \Omega$ [Smets, 1993]:

$$f[A] = \bigcup_{a_i \in A} f[a_i], \quad \forall A \subseteq \Omega, A \neq \emptyset \quad (2.59)$$

⁴If unnormalized belief functions are allowed, the conjunctive rule should be used instead.

Using the disjunctive rule for combining the belief functions $f[a_i]$ in Eq. (2.59) makes sense because A is a union of singletons a_i , meaning that at least one element a_i must hold if A holds. For mass, belief, and plausibility functions, disjunctively combining such singleton-induced belief functions results in the following equations [Smets, 1993]:

$$m[A](B) = \sum_{(\bigcup_{i:a_i \in A} B_i)=B} \prod_{a_i \in A} m[a_i](B_i), \quad (2.60)$$

$$bel[A](B) = \prod_{a_i \in A} bel[a_i](B), \quad (2.61)$$

$$pl[A](B) = 1 - \prod_{a_i \in A} (1 - pl[a_i](B)). \quad (2.62)$$

Another important use of conditional belief functions is that they can sometimes simplify the computation of Dempster's rule of combination. In case it is easier to express a belief function in a conditional form, Dempster's rule for combining two belief functions defined over Θ can be stated in a different way. If f_1 is a normalized belief function with $f \in \{m, bel, pl, q\}$ and m_2 is a normalized mass function, then the combination $f_{1 \oplus 2}$ according to Dempster's rule can be computed by [Dubois and Prade, 1986a]

$$f_{1 \oplus 2}(A) = \eta \sum_{B \subseteq \Theta, B \neq \emptyset} f_1^*[B](A) m_2(B) \quad \forall A \subseteq \Theta, \quad (2.63)$$

$$f_1^*[B](A) = f_1[B](A) pl_1(B). \quad (2.64)$$

Eq. (2.63) is essentially a generalization of the law of total probability. It is therefore referred to as the f -total law and it turns out to be very useful in practice. Note that the conditioning of f_1 with B is generally unnormalized ($bel_1^*[B](\Theta) \leq 1$), making the notation a little cumbersome if the final outcome is supposed to be normalized. Because conditional belief functions are defined in terms of Dempster's rule in Eq. (2.54), the associated normalization has to be undone in Eq. (2.64) by multiplying with the normalization constant $pl_1(B)$ introduced in Eq. (2.56) [Smets, 1991]. There are two important cases in which this “denormalization” can be omitted: (i) if all involved belief functions are unnormalized and $f_{1 \odot 2}$ is computed instead of $f_{1 \oplus 2}$ or (ii) if $f_1^*[B]$ is normalized for any B , which is the case if A and B are defined over different frames of discernment.

2.5. Distinctness and Independence

All combination rules presented in Sect. 2.2 require the underlying pieces of evidence to be “distinct” (except for the cautious rule). Unfortunately, the notion

of distinctness was never defined in a formal way for non-Dempsterian interpretations of belief functions [Shafer and Tversky, 1985]. A detailed discussion of this problem can be found in [Smets, 1992a]. For the special case of Bayesian belief functions, distinctness reduces to probabilistic conditional independence, which can be seen in Eq. (2.35) [Smets, 2007]. For general belief functions, the following definition, which works well in practice and is shared by many works, is adopted (e.g., [Smets, 1998b]):

Definition. *Two pieces of evidence e_1, e_2 are said to be distinct if and only if learning that e_2 is true does not influence the belief induced by e_1 and vice versa.*

Regarding the combination process, this essentially means that no piece of evidence should be counted twice. When considering multiple variables, the notion of distinctness is usually not sufficient. In [Ben Yaghlane et al., 2002a] and [Ben Yaghlane et al., 2002b], the case of multiple variables is considered and the concepts of *irrelevance* and *non-interactivity* are introduced. Let X, Y be two evidential variables with frames of discernment Θ_X, Θ_Y and let f_{Θ_X} be a belief function over Θ_X with $f_{\Theta_X} \in \{m, bel, pl, q\}$. Variable Y is said to be *irrelevant* with respect to X if and only if

$$f_{\Theta_X}[Y] = f_{\Theta_X}, \quad \forall Y \subseteq \Theta_Y. \quad (2.65)$$

The concept of irrelevance is similar to distinctness because Y does not influence the belief in X and vice versa. In particular, irrelevance is a property that can usually be defined by an expert.

In contrast, the concept of non-interactivity concerns the decomposability of belief functions. Variables X and Y are said to be *non-interactive* if and only if the joint belief function $m_{\Theta_X \times \Theta_Y}$ can be constructed from the marginal belief functions over Θ_X and Θ_Y using Dempster's rule.⁵

$$m_{\Theta_X \times \Theta_Y} = m_{\Theta_X} \oplus m_{\Theta_Y} \quad (2.66)$$

For the conditional case with an additional variable Z , non-interactivity of X and Y given z with $z \in \Theta_Z$ is defined as

$$m_{\Theta_X \times \Theta_Y}[z] = m_{\Theta_X}[z] \oplus m_{\Theta_Y}[z], \quad \forall z \in \Theta_Z. \quad (2.67)$$

Non-interactivity implies irrelevance though the opposite is not generally true because non-interactivity additionally requires the preservation of irrelevance under Dempster's rule [Ben Yaghlane et al., 2002a]. The concepts of irrelevance and non-interactivity are equivalent in probability theory where both reduce to the notion of conditional independence.

⁵For unnormalized belief functions, non-interactivity can be defined in terms of the conjunctive rule, though in this case, an additional normalization constant is required [Ben Yaghlane et al., 2002b].

Non-interactivity is equivalent to the concept of *evidential independence* introduced in [Shafer, 1976, chapter 7]. In the same work, the weaker notion of *cognitive independence* is also introduced. Evidential independence implies cognitive independence but not vice versa. Variables X and Y are said to be *cognitively independent* if and only if the joint plausibility function $pl(X, Y)$ can be factorized into the marginal plausibility functions $pl(X)$ and $pl(Y)$.

$$pl(X, Y) = pl(X) pl(Y), \quad \forall X \subseteq \Theta_X, Y \subseteq \Theta_Y. \quad (2.68)$$

If there is a third variable Z , variables X and Y are said to be *conditionally cognitively independent* given z if and only if [Smets, 1993]

$$pl[z](X, Y) = pl[z](X) pl[z](Y), \quad \forall X \subseteq \Theta_X, \forall Y \subseteq \Theta_Y, z \in \Theta_Z. \quad (2.69)$$

Note that for cases where the distinction between non-interactivity, irrelevance, and cognitive independence is not essential, the more-common term “conditional independence” is usually used.

2.6. Generalized Bayesian Theorem

The introduction of the generalized Bayesian theorem by Smets in [Smets, 1993] marks one of the major theoretical results for belief function theory. Without it, belief function theory could hardly be considered a generalization of Bayesian probability theory because there would be no way of handling likelihoods. Let X and Y be evidential variables with separate frames of discernment Θ_X and Θ_Y . Like the name “generalized Bayesian theorem” suggests, it allows constructing a belief function over Θ_X from belief functions over Θ_Y conditioned on elements from Θ_X . Let the singleton-conditioned plausibilities $pl[x](Y)$ with $x \in \Theta_X, Y \subseteq \Theta_Y$ be all the information that is available ($pl[x](Y)$ is referred to as the *likelihood* of x). Assuming each $pl[x](Y)$ results from distinct evidence, the belief function over Θ_X can then be computed in the following way (the equation for $m[Y](X)$ is given in [Delmotte and Smets, 2004], all others can be found in [Smets, 1993]):

$$m[Y](X) = \eta \prod_{x \in X} pl[x](Y) \prod_{x \in \bar{X}} (1 - pl[x](Y)), \quad (2.70)$$

$$bel[Y](X) = \eta \prod_{x \in \bar{X}} bel[x](\bar{Y}) - \eta \prod_{x \in \Theta_X} bel[x](\bar{Y}), \quad (2.71)$$

$$pl[Y](X) = \eta (1 - \prod_{x \in X} (1 - pl[x](Y))), \quad (2.72)$$

$$q[Y](X) = \eta \prod_{x \in X} pl[x](Y), \quad (2.73)$$

$$\eta^{-1} = 1 - \prod_{x \in \Theta_X} (1 - pl[x](Y)), \quad (2.74)$$

$$\forall X \subseteq \Theta_X, Y \subseteq \Theta_Y, X \neq \emptyset, Y \neq \emptyset.$$

These are the normalized versions of the generalized Bayesian theorem. For the unnormalized versions, the normalization constant η is simply omitted. Note that there is no prior over Θ_X in these equations like in the classical Bayesian theorem. Thus, if a prior is not available, it can be ignored (which is not possible with the classical Bayesian theorem). However, in case there is a prior, it is combined with the resulting belief function using Dempster's rule (or the conjunctive rule, depending on whether normalization is desired). Let e_0 denote prior evidence inducing a belief over Θ_X and let $m_{\Theta_X}[Y]$ denote the belief function computed from $pl[x](Y)$ according to Eq. (2.70). The belief function $m_{\Theta_X}[Y, e_0]$ reflecting all the available evidence is then given by

$$m_{\Theta_X}[Y, e_0] = m_{\Theta_X}[Y] \oplus m_{\Theta_X}[e_0]. \quad (2.75)$$

The generalized Bayesian theorem is particularly useful if there is a variable X representing a “cause” and a set of variables Y_i with $1 \leq i \leq n$ representing various “effects”. Just like the classical Bayesian theorem, the generalized Bayesian theorem allows computing the belief over possible causes given the observed effects. In order for the generalized Bayesian theorem to be applicable, the distributions $pl[x](Y_{1:n})$ over the effects need to be independent from each other for each context $x \in \Theta_X$. If furthermore conditional cognitive independence holds between the effects given a common cause, then each cause-effect relation can be described by a separate model $pl[x](Y_i)$. The belief over X can then simply be computed from a set of observed effects $y_{1:n}$ using Eq. (2.70) where the joint effect plausibility is factorized due to conditional cognitive independence.

$$m[y_{1:n}](X) \stackrel{(2.70)}{=} \eta \prod_{x \in X} pl[x](y_{1:n}) \prod_{x \in \bar{X}} (1 - pl[x](y_{1:n})) \quad (2.76)$$

$$\stackrel{(2.69)}{=} \eta \prod_{x \in X} \prod_{i=1}^n pl[x](y_i) \prod_{x \in \bar{X}} (1 - \prod_{i=1}^n pl[x](y_i)) \quad (2.77)$$

Note that this is equivalent to independently computing the belief $m_{\Theta_X}[y_i]$ for each observation y_i using Eq. (2.70) and then combining the belief functions using Dempster's rule [Smets, 1993]).

$$m_{\Theta_X}[y_{1:n}] = \bigoplus_{i=1}^n m_{\Theta_X}[y_i] \quad (2.78)$$

The name “generalized Bayesian theorem” stems from the fact that it reduces to the classical Bayesian theorem if the conditional plausibilities are probability functions and if there is furthermore a Bayesian prior over Θ_X . By assuming $pl[x](y) = P(y|x)$ and $m[e_0](x) = P(x)$ where e_0 represents the prior evidence for x , the classical Bayesian theorem for the posterior $P(x|y)$ can be recovered in the following way (the right hand side shows the respective equations required for each transformation):

$$\begin{aligned} P(x|y) &= q[y, e_0](x) & (\text{Eq. (2.29)}) & (2.79) \end{aligned}$$

$$\propto q[y](x) q[e_0](x) \quad (\text{Eq. (2.75) and (2.36)}) \quad (2.80)$$

$$\propto pl[x](y) q[e_0](x) \quad (\text{Eq. (2.73)}) \quad (2.81)$$

$$= P(y|x) P(x) \quad (\text{assumption}) \quad (2.82)$$

2.7. Pignistic Transformation

In order to make decisions based on belief functions, Smets argues that beliefs first need to be transformed to probabilities [Smets, 2005b, Smets, 2002]. This reflects the distinction between the credal and the pignistic level within the TBM framework where the pignistic level is used for decision making. Only at the pignistic level is it possible to compute an expected value of a utility function, which is the basis for rational decision making.

Transforming a belief function into a pignistic probability function is done via the *pignistic transformation*. Let m denote a mass function with frame of discernment Θ and let $BetP_m$ denote the corresponding pignistic probability function. The pignistic transformation of m into $BetP_m$ is defined as

$$BetP_m(a) = \sum_{A \ni a} \frac{m(A)}{|A| (1 - m(\emptyset))}, \quad \forall a \in \Theta. \quad (2.83)$$

The normalization constant $1 - m(\emptyset)$ can be omitted if m is normalized. Essentially, this transformation causes all mass values assigned to focal sets A with $|A| > 1$ to be evenly distributed among the elements $a \in A$. For example, a vacuous belief function would be transformed into a uniform probability distribution.

(*Remark: If a piece of evidence e induces a belief function $m[e]$, the notation $BetP[e]$ is used to denote the pignistic transformation of $m[e]$.)*

The pignistic transformation can be used to compute the expected value of a real-valued random variable. Let X be a random variable mapping elements from the finite set Θ to the reals \mathbb{R} . Let $BetP_m$ be the pignistic transformation of a mass function m defined over Θ . The expected value $E(X)$ is then defined as [Smets, 2005b]

$$E(X) = \sum_{\theta \in \Theta} X(\theta) BetP_m(\theta). \quad (2.84)$$

2.8. Uncertainty Measures

Both on theoretical and practical grounds, being able to quantify the amount of uncertainty associated with a belief function is useful. For example, in Chap. 5,

a system is presented which actively seeks to minimize uncertainty by selecting actions with the highest expected information gain. Because this system is based on belief functions, it requires a way of measuring the amount of uncertainty associated with a belief function.

For probability functions, the Shannon entropy [Shannon, 1948] constitutes a commonly accepted measure of uncertainty. Let P denote the probability function associated with a discrete random variable X defined over the sample space Θ . The Shannon entropy H is defined as the expected value of the self-information $-\log_b P(x)$ of each event $x \in \Theta$.

$$H(P) = - \sum_{x \in \Theta} P(x) \log_b P(x) \quad (2.85)$$

Because information is usually measured in bits, $b = 2$ is the most common choice. In case there is no uncertainty ($P(x) = 1$ for some $x \in \Theta$), the Shannon entropy is 0. The Shannon entropy increases the more uniform the distribution P becomes and it reaches its maximum for an entirely uniform distribution, which represents a state of total uncertainty in the Bayesian framework.

Unfortunately, there is no commonly accepted measure of uncertainty for general belief functions, despite many attempts to generalize the concept of Shannon entropy (see [Klir and Smith, 2001, Klir, 2004, Klir, 1999, Pal et al., 1992] for overviews). For belief functions, there are two kinds of uncertainty: *non-specificity* and *conflict*.⁶ *Non-specificity* (also called ignorance) refers to the fact that hypotheses are sets with arbitrary cardinality and thus carry uncertainty in themselves (which element in a focal set is true?). This kind of uncertainty is captured by Hartley's measure of non-specificity HL [Hartley, 1928], which quantifies the uncertainty of a focal set in terms of its cardinality. The generalized Hartley measure GH [Klir, 2005] quantifies the overall amount of non-specificity associated with a mass function m .

$$HL(A) = \log_b |A| \quad (2.86)$$

$$GH(m) = \sum_{A \in \mathcal{F}_m} m(A) \log_b |A| \quad (2.87)$$

In contrast, *conflict* is the uncertainty found in classical probability theory with mutually exclusive (i.e., conflicting) events and it is adequately captured by the Shannon entropy.

An uncertainty measure for belief functions should satisfy a number of requirements [Klir, 2005] and it should reflect both non-specificity and conflict. Up to this point, only one measure satisfying all the desired requirements has been found (most measures fail to satisfy the property of sub-additivity). This measure is called *Aggregate Uncertainty* (AU). Let f be a belief function with

⁶Not to be confused with the conflict between different belief function as measured by the weight of conflict Con .

$f \in \{m, bel, pl, q\}$, then AU is defined as the maximum Shannon entropy over the set $\mathbb{P}(f)$ of all probability functions *compatible* (see Eq. (2.9)) with f .

$$AU(f) = \max_{P \in \mathbb{P}(f)} H(P) \quad (2.88)$$

The measure's name results from the fact that it captures non-specificity and conflict in an aggregated fashion. Furthermore, it reduces to Shannon entropy for the special case of a Bayesian belief function. Even though computing the AU measure requires finding the solution to a non-linear optimization problem, there exists an “efficient” algorithm with worst-case time complexity $O(|\mathcal{P}(\Theta)|^2)$ where Θ denotes the frame of discernment of f [Harmanec, 1997]. It should be noted that AU is also an appealing solution in the context of other uncertainty frameworks (e.g., possibility theory).

However, there are also a number of problems with the AU measure. First, it is insensitive to strong belief changes because of the max operation (see [Klir, 2004] for an example). Second, despite the availability of an efficient algorithm, it is computationally expensive, which is particularly problematic if it has to be computed many times. Third, its adequacy is questionable when belief functions are not interpreted as defining lower and upper probability bounds.

For these reasons (in particular the first two), the AU measure is not used in this work. Instead, a simpler measure based on the pignistic transformation is proposed, which is referred to here as *pignistic entropy* [Reineking, 2008]. Like the name suggests, the pignistic entropy H_{BetP} is computed by first applying the pignistic transformation to a belief function and then computing the Shannon entropy of the resulting probability distribution. The pignistic entropy of a mass function m is defined as

$$H_{BetP}(m) = H(BetP_m). \quad (2.89)$$

This measure is justified in particular in a classification context where the pignistic transformation is applied in order to determine the most probable class (which is the case in Chap. 5). In addition, it can be computed much more efficiently with worst-case time complexity $O(|\mathcal{P}(\Theta)|)$, which is a result of computing the pignistic transformation.

3

Evidential Particle Filtering

3.1. Introduction

In this chapter, a particle filter algorithm is derived within the belief function framework. The main results presented in this chapter were originally proposed in [Reineking, 2011]. Some preliminary work regarding the temporal updating of belief functions was presented in [Reineking, 2008].

Particle filtering is one of the most widely-applied probabilistic techniques for estimating the state of a dynamical system where state transitions and observations are subject to uncertainty [Doucet et al., 2001]. Being a Monte-Carlo approach, a finite number of samples (or particles) is used to approximate the probability (density) distribution of the current state. The state is usually described by a low-dimensional vector because the number of samples required for accurately approximating the underlying distribution tends to grow exponentially with the number of dimensions. Especially in robotics, particle filters have been extremely successful for problems like localization [Thrun et al., 2001] and SLAM [Montemerlo et al., 2002], but they are also commonly used in other domains like computer vision [Isard and Blake, 1998].

Compared to Kalman filters [Kalman, 1960], the most popular alternative approach for state estimation, particle filters provide a number of important advantages. In Kalman filters, the distribution of the current state is modeled as a Gaussian and both state transitions and observations are modeled as linear functions with additive Gaussian noise. While the linearity requirement can be relaxed (using extended or unscented Kalman filters [Ljung, 1979, Wan and van der Merwe, 2000]), the class of distributions that can be expressed using Kalman filters is much more restricted compared to particle filters. In particular, particle filters are able to handle multi-modal distributions well and can

also be applied to discrete state variables. Because of these reasons, particle filters have become the standard solution for estimation problems in many domains.

For situations where the state prior, transitions, or observations can be more accurately described by belief functions opposed to probability functions, being able to perform particle filtering based on belief functions would be very useful. The approach presented in this chapter does exactly this by generalizing the concept of particle filtering in the belief function framework. The two major contributions of the work presented in this chapter are:

- Derivation of recursive update equations based on belief functions which generalize Bayesian filtering.
- Efficient approximations of the solutions to these equations based on an evidential particle filter algorithm.

The belief about the current state, transitions, and observations are all described by belief functions in this case, allowing the use of richer uncertainty models capable of expressing partial or total ignorance when appropriate. In order to make the computations tractable, the belief about the current state is approximated by a finite number of samples. This reduces the exponential complexity of the exact solution to a complexity which is linear with respect to the size of the state space. It should be noted that, throughout this chapter, the state is assumed to be a discrete variable. In contrast, observations can be discrete or continuous (handling continuous measurements is described in [Smets, 2005a]). A discussion of extending the presented approach to continuous state variables can be found at the end of the chapter.

The remainder of this chapter is structured as follows. The next section provides a brief overview of related work. In Sect. 3.3, a belief function filter and its recursive update equations are derived. An evidential particle filter algorithm, a Monte-Carlo solution to the update equations, is presented in Sect. 3.4 along with a detailed derivation. The computational complexity of the algorithm and the corresponding approximation error are analyzed in Sect. 3.5. In Sect. 3.6, the algorithm is applied to a bearings-only tracking problem where the tracking error of the evidential solution is compared to the performance of a corresponding probabilistic solution. In the final section, the presented approach and possible extensions are discussed.

3.2. Related Work

The work related to the approach presented in this chapter can be roughly divided into two areas. The first is concerned with modeling temporal processes using belief functions. The second is concerned with Monte-Carlo approaches for efficiently combining belief functions.

3.2.1. Temporal Updating of Belief Functions

In the context of filtering based on belief functions, a solution to the problem of joint tracking and classification in the TBM framework is proposed in [Smets and Ristic, 2007, Smets and Ristic, 2004]. The authors derive a Kalman filter algorithm where the underlying belief functions are implicitly represented by their pignistic transformations which are Gaussians. The evidential Kalman filter is used to track an object and estimate its class based on the observed motion behavior. Due to the evidential representation, it is possible to dissociate the motion behavior from the class, resulting in a higher classification accuracy compared to a Bayesian solution.

There are several works about filtering based on probabilistic dynamics and evidential processing of observations. In [Muñoz-Salinas et al., 2009], a person is tracked by multiple cameras where belief functions are used to handle the problem of the person being out of sight. Similarly, in [Klein et al., 2010], a particle filter performs visual tracking and different belief function combination rules are used to fuse information sources based on their reliability and precision.

In [Ramasso et al., 2007a], hidden Markov models are generalized within the TBM framework. Beliefs are represented by commonality functions here which simplifies some of the computations. Among other things, an evidential version of the Viterbi algorithm is proposed and an application to human motion analysis is described (cf. [Ramasso et al., 2007b]).

3.2.2. Monte-Carlo Approximations of Belief Functions

When dealing with large frames of discernment, exact belief function inference becomes infeasible and approximate approaches are required. The use of Monte-Carlo methods is an obvious choice in this case, which is why several sampling-based inference approaches for belief functions have been proposed over time [Wilson, 2000, Moral and Wilson, 1996, Moral and Salmerón, 1999, Kreinovich et al., 1994]. Despite the success of Monte-Carlo approaches in many fields, their overall popularity in the context of belief function theory is still limited.

Most Monte-Carlo approaches for belief functions aim to approximate the belief resulting from combining two or more belief functions using Dempster's rule of combination. For Dempster's rule, the normalization is particularly problematic with respect to sampling. Without normalization (e.g., for the conjunctive rule of combination) or when conflict is low, a simple Monte-Carlo algorithm could independently draw samples from each mass function and build the intersection of these samples. However, for highly conflicting belief functions, most intersections would be empty and thus invalid if no mass on \emptyset is allowed. Therefore, such a simple Monte-Carlo algorithm would either yield poor approximations (with few valid samples) or it would be very inefficient (when repeating the sampling process until enough non-empty intersections are found).

For this reason, two Monte-Carlo methods have been proposed which are able to handle highly conflicting evidence when approximating Dempster's rule. The first method is based on Markov chain Monte-Carlo [Wilson and Moral, 1996] where a Markov chain is used in order to draw samples from the underlying belief distribution. While yielding good approximations, this method is not suited for particle filtering because it requires multiple runs in order to produce accurate results, which is not acceptable for an online particle filter algorithm. The second method is importance sampling [Moral and Wilson, 1996]. Here, samples are drawn from a different distribution (i.e., one which makes empty intersections impossible), and importance weights are used to account for the difference between the true distribution and the sampled one. A detailed description of importance sampling is given in Sect. 3.4.2. Note that importance sampling is also used in most probabilistic particle filter algorithms.

3.3. Evidential Filtering

In this section, the equations for filtering based on belief functions are derived. In addition, it is shown that these equations reduce to Bayesian filtering [Thrun et al., 2005, chapter 2] if the prior and the underlying transition and observation models are Bayesian belief functions.

Let Θ_t denote the discrete state space for state $x_t \in \Theta_t$ at time t .¹ Let Ω_t denote the discrete or real-valued observation space for observation $z_t \in \Omega_t$ at time t (e.g., z_t could be a sensor measurement). Let $z_{0:t}$ denote the sequence of all observations obtained over time (short for z_0, \dots, z_t). Without a loss of generality, it is assumed that state transitions and observations occur in alternating order, i.e., after each state transition from x_{t-1} to x_t , an observation z_t is recorded. This corresponds to the dynamic belief network shown in Fig. 3.1 (belief networks are a generalization of Bayesian networks [Xu and Smets, 1996]). The aim of filtering is to compute the belief about the current state x_t given all previous observations $z_{0:t}$. The underlying process is assumed to satisfy the (first-order) Markov property, i.e., given the state at time t , all prior information including past states and observations is irrelevant for future states and observations (a generalization to higher-order Markov properties is straightforward). This also means that each observation is assumed to be conditionally independent from all other observations given the state in which it occurred.

The goal of evidential filtering is to compute the mass function $m_{\Theta_t}[z_{0:t}]$ which expresses the belief about the current state $X_t \subseteq \Theta_t$ given all observations $z_{0:t}$ up to time t .² The reason for using a mass function to represent the belief about

¹The state and observation spaces are usually time-invariant, however, time indices are kept for clarity.

²The subscript Θ_t is used to make the space over which the belief function is defined explicit. In case the space is clear from the variables, the subscript is omitted in order to keep notation concise (e.g., $m(X_t)$ instead of $m_{\Theta_t}(X_t)$).

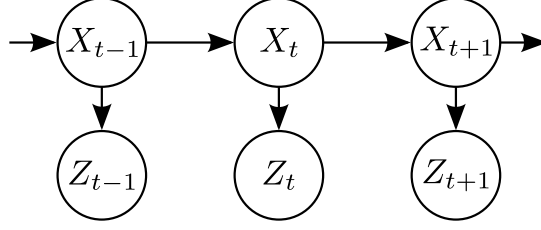


Figure 3.1.: Dynamic belief network showing how evidential state and observation variables relate to each other. Each observation Z_i depends on the respective state X_i while the sequence of all states forms a first-order Markov chain. Because states cannot be observed directly, the belief network describes a Hidden Markov process.

the current state is that a mass function can be interpreted as a probability distribution over the power set of the state space, which can be approximated using a finite number of samples. In contrast to equivalent representations like *bel*, *pl*, or *q*, mass functions are generally also much sparser and therefore easier to approximate if the number of focal sets is limited. Even if the number of focal sets is large, if many focal sets have mass values close to 0, it is still possible to obtain a good approximation using a limited number of samples.

Throughout this chapter, belief functions are assumed to be normalized. The reason for this assumption is the fact that each new observation requires a combination of the current state belief with the new observation-induced belief, usually involving a non-zero weight of conflict. If this combination is unnormalized, the mass associated with \emptyset would converge to 1 over time. In this case, the mass on \emptyset would mainly serve as an indicator of how many observations the filter algorithm has processed, which would not be very useful. In addition, approximating such an unnormalized mass function using a set of samples would be problematic because, after a while, almost all samples would represent \emptyset , leaving only very few samples for the “interesting” mass values associated with the remaining focal sets.

Like with Bayesian filtering, there are two situations in which the belief about the current state needs to be updated: state transitions and new observations. To keep nomenclature consistent with Bayesian filtering, the update in case of state transitions is referred to as the *prediction step* while the update due to a new observation is referred to as the *correction step*. These two steps allow the belief about the current state to be updated recursively over time. Aside from the models describing state transitions and observations, only the initial belief m_{Θ_0} has to be specified. Unless there is evidence indicating otherwise, this belief is assumed to be vacuous. In contrast, when using a Bayesian approach, this state of ignorance would have to be modeled as a uniform distribution, and sampling-based approximations of uniform distributions typically result in considerable errors because the samples have to cover the entire state space.

3.3.1. Prediction Step

In the prediction step, the mass function $m_{\Theta_t}[z_{0:t-1}]$ is computed which represents the belief about the current state X_t while reflecting all observations up to time $t-1$. Note that $m_{\Theta_t}[z_{0:t-1}]$ is referred to as the *proposal distribution*. This mass function can be obtained by applying Dempster's rule of combination to the prior belief $m_{\Theta_{t-1}}[z_{0:t-1}]$ and the transition belief $m_{\Theta_{t-1} \times \Theta_t}$ defined over the joint space $\Theta_{t-1} \times \Theta_t$. To do so, the prior belief first needs to be extended to the joint space and the result of the combination is marginalized over Θ_t afterwards.

$$m_{\Theta_t}[z_{0:t-1}] = (m_{\Theta_{t-1} \times \Theta_t} \oplus m_{\Theta_{t-1}}[z_{0:t-1}])^{\uparrow \Theta_{t-1} \times \Theta_t} \downarrow_{\Theta_t} \quad (3.1)$$

Using the f -total law defined by Eq. (2.63), this combination can be expressed more elegantly using conditional mass functions by conditioning with every possible prior focal set $X_{t-1} \subseteq \Theta_{t-1}$.

$$m[z_{0:t-1}](X_t) \stackrel{(2.63)}{=} \eta \sum_{X_{t-1} \subseteq \Theta_{t-1}} pl_{\Theta_{t-1} \times \Theta_t}^{\downarrow \Theta_{t-1}}(X_{t-1}) m[X_{t-1}](X_t) m[z_{0:t-1}](X_{t-1}) \quad (3.2)$$

The conditional mass function $m_{\Theta_t}[X_{t-1}]$ is called the *transition model*. It describes how the state is expected to change over time.³ The normalization constant η results from the application of Dempster's rule while $pl_{\Theta_{t-1} \times \Theta_t}^{\downarrow \Theta_{t-1}}(X_{t-1})$ is the normalization constant associated with the f -total law. Below it is shown that both normalization constants can be omitted because they turn out to be 1.

The transition model can either be directly specified or it can be constructed using the disjunctive rule of combination. In the latter case, it is assumed that all the available information about state transitions is represented by the set of singleton-conditioned distributions $\{m_{\Theta_t}[x_{t-1}] | x_{t-1} \in \Theta_{t-1}\}$. Assuming all these distributions are induced by distinct evidence, the disjunctive rule of combination can be used to build the transition model conditioned on arbitrary sets $X_{t-1} \subseteq \Theta_{t-1}$ of prior states (see Sect. 2.4).

$$m_{\Theta_t}[X_{t-1}] \stackrel{(2.59)}{=} \bigcup_{x_{t-1} \in X_{t-1}} m_{\Theta_t}[x_{t-1}], \quad \forall X_{t-1} \subseteq \Theta_{t-1}, X_{t-1} \neq \emptyset \quad (3.3)$$

$$m[X_{t-1}](X_t) \stackrel{(2.60)}{=} \sum_{\substack{X_{t-1} \subseteq \Theta_{t-1} \\ i: x_{t-1,i} \in X_{t-1}}} \prod_{x_{t-1,i} \in X_{t-1}} m[x_{t-1,i}](X_{t,i}) \quad (3.4)$$

When constructing the transition model in such a way, it is vacuous over Θ_{t-1} . More specifically, the disjunctive rule of combination in Eq. (3.3) causes

³In the context of robotics, there is usually additional information like odometry describing state transitions. Here, such information is not considered but in case it is available, it can be easily incorporated into the transition model.

all elements of X_{t-1} to be part of every focal set (over the joint space). Thus, when marginalizing over Θ_{t-1} , the transition model is vacuous with

$$m_{\Theta_{t-1} \times \Theta_t}^{\downarrow \Theta_{t-1}}(\Theta_{t-1}) = 1. \quad (3.5)$$

This means that $pl_{\Theta_{t-1} \times \Theta_t}^{\downarrow \Theta_{t-1}}(X_{t-1}) = 1$ for any $X_{t-1} \subseteq \Theta_{t-1}, X_{t-1} \neq \emptyset$. For the same reason, one has $\eta = 1$ in Eq. (3.2) because the transition model is vacuous over Θ_{t-1} while the prior belief $m_{\Theta_{t-1}}[z_{0:t-1}]$ is by definition vacuous over Θ_t , i.e., there are no empty intersections in the resulting combination. The complete prediction step results from plugging Eq. (3.4) into Eq. (3.2) and removing the normalization constants.

$$m[z_{0:t-1}](X_t) = \sum_{X_{t-1} \subseteq \Theta_{t-1}} m[z_{0:t-1}](X_{t-1}) m[X_{t-1}](X_t) \quad (3.6)$$

$$= \sum_{X_{t-1} \subseteq \Theta_{t-1}} m[z_{0:t-1}](X_{t-1}) \sum_{\substack{\bigcup_{i: x_{t-1,i} \in X_{t-1}} X_{t,i} = X_t}} \prod_{x_{t-1,i} \in X_{t-1}} m[x_{t-1,i}](X_{t,i}) \quad (3.7)$$

3.3.2. Correction Step

In the correction step, a new observation z_t is incorporated into the proposal distribution $m_{\Theta_t}[z_{0:t-1}]$ computed in the prediction step according to an observation model. The aim is to compute the mass function $m_{\Theta_t}[z_{0:t}]$ reflecting all evidence up to time t . Each observation z_i is assumed to be conditionally independent from all other observations given the corresponding state X_i (more specifically, each observation is *conditionally non-interactive* with respect to all other observations, see Sect. 2.5). Therefore, the belief induced by observation z_t can be combined with the proposal belief using Dempster's rule of combination.

$$m_{\Theta_t}[z_{0:t}] \stackrel{(2.67)}{=} m_{\Theta_t}[z_{0:t-1}] \oplus m_{\Theta_t}[z_t] \quad (3.8)$$

In case the mass function $m_{\Theta_t}[z_t]$ induced by observation z_t can be directly specified, it could simply be combined with the proposal distribution and there would be nothing else to do. More commonly though, there is a generative observation model providing a belief distribution over the observation space Ω_t given a particular state $x_t \in \Theta_t$. Similar to how the transition model is created from a set of mass functions for each singleton x_{t-1} in Eq. (3.3), the observation model is assumed to result from the knowledge of a set of likelihoods $pl_{\Omega_t}[x_t]$ for each $x_t \in \Theta_t$. The generalized Bayesian theorem can then be applied to construct $m_{\Theta_t}[z_t]$ from this set of likelihood functions.

$$m[z_t](X_t) \stackrel{(2.70)}{\propto} \prod_{x_t \in X_t} pl[x_t](z_t) \prod_{x_t \in \bar{X}_t} (1 - pl[x_t](z_t)), \quad \forall X_t \subseteq \Theta_t, X_t \neq \emptyset \quad (3.9)$$

The complete correction step is thus given by plugging Eq. (3.9) into Eq. (3.8) and writing out Dempster's rule of combination defined by Eq. (2.30) in full.

$$m[z_{0:t}](X_t) = \eta \sum_{X'_t \cap X''_t = X_t} m[z_{0:t-1}](X'_t) \prod_{x_t \in X''_t} pl[x_t](z_t) \prod_{x_t \in \bar{X}''_t} (1 - pl[x_t](z_t)) \quad (3.10)$$

Eq. (3.7) and (3.10) are the two equations underlying evidential filtering. Applied in alternating order, these two equations make it possible to recursively update the belief about the state of a dynamical system.

3.3.3. Reduction to Bayesian Filtering

Below it is proved that the derived evidential filtering equations reduce to Bayesian filtering equations if the transition model, the observation model, and the prior are probability functions. This shows that evidential filtering indeed constitutes a generalization of Bayesian filtering.

Prediction Step

The Bayesian solution to the prediction step consists of an application of the total law of probability to the prior $P(x_{t-1}|z_{0:t-1})$. By additionally exploiting the Markov property of the underlying process, the proposal distribution $P(x_t|z_{0:t-1})$ can be computed from the prior and the transition model $P(x_t|x_{t-1})$.

$$P(x_t|z_{0:t-1}) = \sum_{x_{t-1} \in \Theta_{t-1}} P(x_{t-1}|z_{0:t-1}) P(x_t|x_{t-1}) \quad (3.11)$$

Assuming $m[z_{0:t-1}](x_{t-1}) = P(x_{t-1}|z_{0:t-1})$ for the prior and $m[x_{t-1}](x_t) = P(x_t|x_{t-1})$ for the transition model for all x_{t-1} and x_t , the evidential prediction step defined by Eq. (3.6) reduces to Eq. (3.11) because all focal sets are singletons.

$$\begin{aligned} & m[z_{0:t-1}](x_t) \\ &= \sum_{X_{t-1} \subseteq \Theta_{t-1}} m[z_{0:t-1}](X_{t-1}) m[X_{t-1}](x_t) \quad (\text{Eq. (3.6)}) \end{aligned} \quad (3.12)$$

$$= \sum_{x_{t-1} \in \Theta_{t-1}} P(x_{t-1}|z_{0:t-1}) P(x_t|x_{t-1}) \quad (\text{assumption}) \quad (3.13)$$

■

Correction Step

The Bayesian correction step results from an application of the classical Bayesian theorem along with a conditional independence assumption regarding the observations given the current state.

$$P(x_t|z_{0:t}) \propto P(z_t|x_t) P(x_t|z_{0:t-1}) \quad (3.14)$$

By assuming $m[z_{0:t-1}](x_t) = P(x_t|z_{0:t-1})$ for the proposal distribution and $pl[x_t](z_t) = P(z_t|x_t)$ for the observation model, the evidential correction step reduces to Eq. (3.14).

$$m[z_{0:t}](x_t) = (m_{\Theta_t}[z_{0:t-1}] \oplus m_{\Theta_t}[z_t])(x_t) \quad (\text{Eq. (3.8)}) \quad (3.15)$$

$$\propto P(x_t|z_{0:t-1}) pl[z_t](x_t) \quad (\text{proof A.1}) \quad (3.16)$$

$$\propto P(x_t|z_{0:t-1}) pl[x_t](z_t) \quad (\text{Eq. (2.72)}) \quad (3.17)$$

$$= P(x_t|z_{0:t-1}) P(z_t|x_t) \quad (\text{assumption}) \quad (3.18)$$

■

3.4. Evidential Particle Filtering

Analyzing the prediction and correction step equations (3.6) and (3.10), it is obvious that the time complexity of computing these equations is (at least) exponential in the worst case because of the summation over all subsets of the state space. Thus even a small state space (e.g., $|\Theta| = 20$) is computationally challenging while larger state spaces (e.g., $|\Theta| = 100$) make the update computationally intractable unless the involved belief functions only contain a small number of focal sets. Usually, the number of focal sets grows exponentially with increasing state space size though, in which case one has to resort to approximate solutions.

In this section, a Monte-Carlo approach is presented which approximates the belief function resulting from evidential filtering. This approximation is based on a fixed-size set \mathcal{X}_t consisting of K samples $X_t^{[k]} \subseteq \Theta_t$ with $1 \leq k \leq K$, which is updated over time. The relative frequency of a hypothesis X_t in the sample set \mathcal{X}_t is an estimate of its true mass value. As the number of samples K goes to infinity, the relative frequency of each hypothesis converges in probability to the true mass value of the hypothesis.

As a result, space and time complexity of representing and combining the corresponding belief functions are reduced from exponential to linear with respect to the size of the state space (with an additional constant factor determined by the number of samples, see Sect. 3.5). The resulting algorithm represents an evidential generalization of a discrete-state probabilistic particle filter. More specifically, the algorithm follows a sequential importance resampling (SIR) scheme [Gordon et al., 1993], where importance sampling is used to efficiently approximate Dempster's rule of combination underlying the correction step.

The remainder of this section is split into four parts. In part one and two, algorithms for approximating the prediction and correction step are presented. In the third part, an algorithm for efficiently sampling from the observation-induced distribution $m_{\Theta_t}[z_t]$ is shown. In the final part, the algorithms are derived in a more rigorous manner and they are shown to correctly approximate the analytical solutions.

Algorithm: Prediction step	
Input: \mathcal{X}_{t-1}	// prior sample set
1 $\hat{\mathcal{X}}_t \leftarrow \emptyset$	// proposal sample set
2 for $k \leftarrow 1$ to K do	
3 $\hat{X}_t^{[k]} \leftarrow \emptyset$	// proposal sample
4 foreach $x_{t-1;i}^{[k]} \in X_{t-1}^{[k]}$ do	
5 sample $\hat{X}_{t;i}^{[k]} \sim m_{\Theta_t}[x_{t-1;i}^{[k]}]$	// transition model
6 $\hat{X}_t^{[k]} \leftarrow \hat{X}_t^{[k]} \cup \hat{X}_{t;i}^{[k]}$	// disjunctive combination
7 end	
8 add $\hat{X}_t^{[k]}$ to $\hat{\mathcal{X}}_t$	
9 end	
10 return $\hat{\mathcal{X}}_t$	

Figure 3.2.: Monte-Carlo approximation of the prediction step. The input \mathcal{X}_{t-1} is a set of samples representing the prior distribution $m_{\Theta_{t-1}}[z_{0:t-1}]$ while the output $\hat{\mathcal{X}}_t$ is a set of samples representing the proposal distribution $m_{\Theta_t}[z_{0:t-1}]$. (Algorithm adopted from [Reineking, 2011].)

3.4.1. Prediction Step

In the prediction step defined by Eq. (3.7), the proposal distribution $m_{\Theta_t}[z_{0:t-1}]$ is computed from the prior distribution $m_{\Theta_{t-1}}[z_{0:t-1}]$. Therefore, the algorithm implementing the prediction step shown in Fig. 3.2 takes as input the sample set \mathcal{X}_{t-1} representing the prior distribution and returns the updated sample set $\hat{\mathcal{X}}_t$ representing the proposal distribution. The proposal sample set is created by transforming each prior sample $X_{t-1}^{[k]} \in \mathcal{X}_{t-1}$ into a proposal sample $\hat{X}_t^{[k]} \in \hat{\mathcal{X}}_t$ by sampling from the transition model $m_{\Theta_t}[X_{t-1}^{[k]}]$ (lines 3–8). Because the transition model is defined as the disjunctive combination of distributions $m_{\Theta_t}[x_{t-1}]$ conditioned by singleton prior states $x_{t-1} \in \Theta_{t-1}$ in Eq. (3.3), a sample is generated independently from each distribution $m_{\Theta_t}[x_{t-1;i}^{[k]}]$ (line 5). Here, the notation $\hat{X}_{t;i}^{[k]} \sim m_{\Theta_t}[x_{t-1;i}^{[k]}]$ indicates that the probability of sample $\hat{X}_{t;i}^{[k]}$ taking on a value \hat{X}_t is equal to the mass $m[x_{t-1;i}^{[k]}](\hat{X}_t)$.

$$P(\hat{X}_{t;i}^{[k]} \leftarrow \hat{X}_t | x_{t-1;i}^{[k]} \leftarrow x_{t-1}) = m[x_{t-1}](\hat{X}_t), \quad \forall \hat{X}_t \subseteq \Theta_t, x_{t-1} \in \Theta_{t-1} \quad (3.19)$$

The union of these singleton-conditioned samples forms the new proposal sample $\hat{X}_t^{[k]}$ (line 6), which corresponds to a disjunctive combination. This entire process is illustrated in Fig. 3.3.

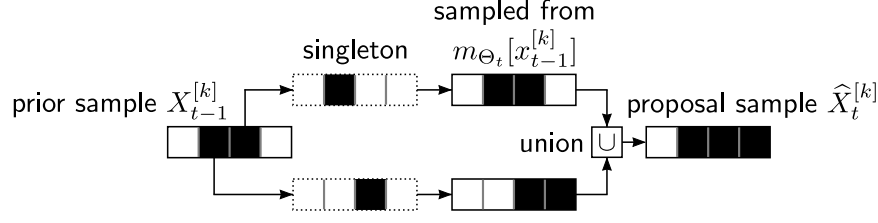


Figure 3.3.: Illustration of how a prior sample $X_{t-1}^{[k]}$ is transformed into a proposal sample $\hat{X}_t^{[k]}$ in the prediction step. Each square represents an element of the state space (in this example, there are only four possible states). For each element of the prior sample, a new sample is drawn from the singleton-conditioned transition model $m_{\Theta_t}[x_{t-1}^{[k]}]$. These samples are then united to form the new proposal sample.

3.4.2. Correction Step

In the correction step defined by Eq. (3.10), the proposal distribution is combined with the observation-induced distribution $m_{\Theta_t}[z_t]$ using Dempster's rule. Because the proposal distribution is already represented as a set of samples $\hat{\mathcal{X}}_t$ according to the prediction algorithm, samples only need to be drawn from $m_{\Theta_t}[z_t]$. A naive Monte-Carlo approach for approximating the combined mass function could consist of independently drawing K samples from $m_{\Theta_t}[z_t]$ and intersecting each one with the corresponding proposal sample, thus creating K new samples representing the combination. The problem with this approach is that many of these intersections would be empty, which is not allowed when using normalized belief functions. Only relying on the non-empty intersections is not an option because the approximation would suffer severely (all intersections could be empty in extreme cases) and repeating the sampling process until K non-empty intersections are found would be very time-consuming. Thus, a more efficient solution is required and this solution is based on *importance sampling* [Moral and Wilson, 1996].

The general idea of importance sampling is to draw samples from a distribution that is easier to sample than the target distribution. The sampling bias caused by the difference between these distributions is accounted for by so-called *importance weights*, which are used to weight each sample from the biased distribution. In addition to weighting, for belief functions where each sample represents a set, the intersections underlying Dempster's rule also have to be computed. The importance weights can either be maintained over time resulting in a *sequential importance sampling* (SIS) particle filter or they can be removed in a resampling step resulting in a *sequential importance resampling* (SIR) particle filter. The algorithm shown in Fig. 3.4 is a Monte-Carlo solution to the correction step and it follows a SIR updating scheme (like most probabilistic particle filter algorithms). The advantage of resampling is that

Algorithm: Correction step	
Input: $\hat{\mathcal{X}}_t, z_t$ // proposal sample set, new observation	
1 $\tilde{\mathcal{X}}_t \leftarrow \emptyset$	// weighted sample set
2 $\mathcal{X}_t \leftarrow \emptyset$	// updated sample set
3 for $k \leftarrow 1$ to K do	
4 $\tilde{X}_t^{[k]} \leftarrow \text{compatible_sample}(z_t, \hat{X}_t^{[k]})$	// sample $\tilde{X}_t^{[k]} \sim m_{\Theta_t}[z_t, \hat{X}_t^{[k]}]$
5 $w_t^{[k]} \leftarrow pl[z_t](\hat{X}_t^{[k]})$	// compute importance weight
6 add $(\tilde{X}_t^{[k]}, w_t^{[k]})$ to $\tilde{\mathcal{X}}_t$	
7 end	
8 for $k \leftarrow 1$ to K do	// importance resampling
9 draw $X_t^{[k]}$ from $\tilde{\mathcal{X}}_t$ with probability $\propto w_t^{[k]}$	
10 add $X_t^{[k]}$ to \mathcal{X}_t	
11 end	
12 return \mathcal{X}_t	

Figure 3.4.: Monte-Carlo approximation of the correction step. The input $\hat{\mathcal{X}}_t$ is a set of samples representing the proposal distribution and z_t is a new observation. The function `compatible_sample` is used to avoid empty intersections during sampling from the observation-induced distribution (the function is defined in Fig. 3.6). The output \mathcal{X}_t is a set of samples representing the updated distribution, which reflects the entire sequence $z_{0:t}$ of observations. (Algorithm adopted from [Reineking, 2011].)

more samples are used to represent focal sets with high mass values whereas, if weights are maintained over time, the sample set tends to degenerate such that most samples represent hypotheses associated with low mass values.

The correction step algorithm implements the combination of the proposal distribution $m_{\Theta_t}[z_{0:t-1}]$ and the observation-induced distribution $m_{\Theta_t}[z_t]$ according to Dempster's rule as defined by Eq. (3.8). Using the idea of importance sampling, for the k -th sample, instead of directly drawing a sample from $m_{\Theta_t}[z_t]$, each sample $\tilde{X}_t^{[k]}$ is drawn from the distribution $m_{\Theta_t}[z_t, \hat{X}_t^{[k]}]$ conditioned by the corresponding proposal sample $\hat{X}_t^{[k]}$. This is done by the function `compatible_sample` (line 4) which is described further below. Drawing a non-empty sample $\tilde{X}_t^{[k]}$ from this conditioned distribution asserts that the intersection with proposal sample $\hat{X}_t^{[k]}$ is never empty. This follows directly from the definition of conditional mass functions in Eq. (2.55) which implies $\tilde{X}_t^{[k]} \subseteq \hat{X}_t^{[k]}$ and thus also

$$\tilde{X}_t^{[k]} \cap \hat{X}_t^{[k]} \neq \emptyset, \quad \forall \hat{X}_t^{[k]}, \tilde{X}_t^{[k]} \subseteq \Theta_t, \hat{X}_t^{[k]} \neq \emptyset, \tilde{X}_t^{[k]} \neq \emptyset. \quad (3.20)$$

In order to correct for the error introduced by biasing the sampling process

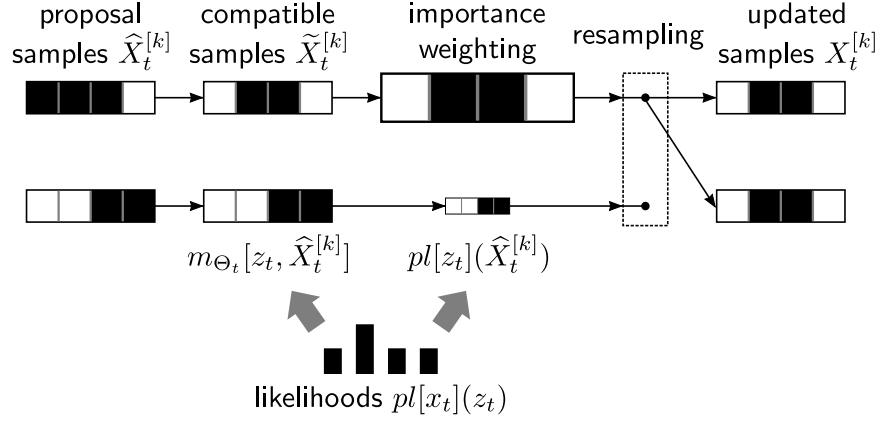


Figure 3.5.: Illustration of how proposal samples $\hat{X}_t^{[k]}$ are updated in the correction step based on observation z_t . The observation induces a likelihood for each singleton state x_t , based on which compatible samples $\tilde{X}_t^{[k]}$ and importance weights $pl[z_t](\hat{X}_t^{[k]})$ are computed using the generalized Bayesian theorem. Finally, resampling is performed according to the importance weights in order to obtain the updated sample set.

with the proposal sample, an importance weight $w_t^{[k]}$ is computed in line 5. The importance weight is defined as the ratio between the target distribution $m_{\Theta_t}[z_t]$ and the sampled distribution $m_{\Theta_t}[z_t, \hat{X}_t^{[k]}]$. It is simply given by the plausibility $pl[z_t](\hat{X}_t^{[k]})$ because that is the normalization constant associated with conditioning by $\hat{X}_t^{[k]}$.

$$w_t^{[k]} = \frac{m[z_t](X_t)}{m[z_t, \hat{X}_t^{[k]}](X_t)} \stackrel{(2.56)}{=} pl[z_t](\hat{X}_t^{[k]}), \quad \forall X_t \subseteq \hat{X}_t^{[k]}, X_t \neq \emptyset \quad (3.21)$$

Note that $m[z_t, \hat{X}_t^{[k]}](X_t)$ is undefined if $pl[z_t](\hat{X}_t^{[k]}) = 0$. This case can be easily handled though by first checking whether $pl[z_t](\hat{X}_t^{[k]}) = 0$, in which case the corresponding proposal sample is ignored (with a weight of 0, the resampling process would ignore the sample anyway).

Usually, the plausibility $pl[z_t](\hat{X}_t^{[k]})$ is expensive to compute for arbitrary belief functions [Wilson, 2000]. However, in this case, $pl[z_t](\hat{X}_t^{[k]})$ is known to result from applying the generalized Bayesian theorem because the corresponding mass function is defined that way in Eq. (3.9). Using the generalized Bayesian theorem for plausibility functions yields the following expression which can be efficiently computed (normalization can be ignored because only weight ratios matter for the resampling process):

$$pl[z_t](\hat{X}_t^{[k]}) \stackrel{(2.72)}{\propto} 1 - \prod_{\hat{x}_t^{[k]} \in \hat{X}_t^{[k]}} (1 - pl[\hat{x}_t^{[k]}](z_t)), \quad \forall \hat{X}_t^{[k]} \subseteq \Theta_t, \hat{X}_t^{[k]} \neq \emptyset. \quad (3.22)$$

Function: compatible_sample	
Input: $z_t, \hat{X}_t^{[k]}$	// new observation, proposal sample
1 $\tilde{X}_t^{[k]} \leftarrow \emptyset$	// sample compatible with $\hat{X}_t^{[k]}$
2 foreach $x_{t,i}^{[k]} \in \hat{X}_t^{[k]}$ do	
3 if $\tilde{X}_t^{[k]} = \emptyset$ then	
4 $\eta_i^{-1} \leftarrow 1 - \prod_{j=i}^{ \hat{X}_t^{[k]} } (1 - pl[x_{t,j}^{[k]}](z_t))$	// normalization
5 else	
6 $\eta_i^{-1} \leftarrow 1$	
7 end	
8 if $\eta_i pl[x_{t,i}^{[k]}](z_t) > r_i$ then	// random number $r_i \in [0, 1)$
9 $\tilde{X}_t^{[k]} \leftarrow \tilde{X}_t^{[k]} \cup \{x_{t,i}^{[k]}\}$	
10 end	
11 end	
12 return $\tilde{X}_t^{[k]}$	

Figure 3.6.: Definition of function `compatible_sample` which generates a non-empty sample from the distribution $m_{\Theta_t}[z_t, \hat{X}_t^{[k]}]$. The input is an observation z_t and a proposal sample $\hat{X}_t^{[k]}$. The output is a randomly drawn sample $\tilde{X}_t^{[k]} \subseteq \hat{X}_t^{[k]}$ with $\tilde{X}_t^{[k]} \neq \emptyset$. (Algorithm adopted from [Reineking, 2011].)

After having computed the importance weight, the tuple $(\tilde{X}_t^{[k]}, w_t^{[k]})$ is added to the temporary weighted sample set $\tilde{\mathcal{X}}_t$ (line 6). In the resampling process (lines 8–11), the final, unweighted sample set \mathcal{X}_t is created by drawing samples with replacement from the weighted set $\tilde{\mathcal{X}}_t$ with a probability proportional to the importance weights. An illustration of how the correction step algorithm works is shown in Fig. 3.5.

3.4.3. Generating Compatible Samples

As described in the previous section, the purpose of function `compatible_sample` is to draw a non-empty sample $\tilde{X}_t^{[k]}$ from the distribution $m_{\Theta_t}[z_t, \hat{X}_t^{[k]}]$. This distribution is computed from the observation likelihoods $pl[x_t](z_t)$ using the generalized Bayesian theorem (see Eq. (3.9)).

$$m[z_t, \hat{X}_t^{[k]}](\tilde{X}_t) \stackrel{(3.9)}{=} \eta \prod_{x_t \in \tilde{X}_t} pl[x_t](z_t) \prod_{x_t \in \tilde{\mathcal{X}}_t} (1 - pl[x_t](z_t)), \quad (3.23)$$

$$\forall \tilde{X}_t \subseteq \hat{X}_t^{[k]}, \tilde{X}_t \neq \emptyset$$

Fig. 3.6 shows an algorithm that implements this sampling process efficiently. Because of $\tilde{X}_t^{[k]} \subseteq \hat{X}_t^{[k]}$, only elements of the proposal sample $\hat{X}_t^{[k]}$ are considered

for generating $\tilde{X}_t^{[k]}$ (line 2). The sample $\tilde{X}_t^{[k]}$ can be interpreted as a random binary vector v of length $|\hat{X}_t^{[k]}|$ where a 1 at the i -th component represents inclusion of singleton $x_{t,i}^{[k]}$ into the sample $\tilde{X}_t^{[k]}$ while a 0 represents exclusion $x_{t,i}^{[k]} \notin \tilde{X}_t^{[k]}$. If the components of v are assumed to be independent, the corresponding probability distribution can be factorized in the following way:

$$P(v) = \prod_{i \in \{j | v_j=1\}} P(v_i) \prod_{i \in \{j | v_j=0\}} (1 - P(v_i)). \quad (3.24)$$

Setting the inclusion probability equal to the likelihood of $x_{t,i}^{[k]}$ with $P(v_i) = pl[x_{t,i}^{[k]}](z_t)$ makes Eq. (3.24) equivalent to Eq. (3.23) apart from the constraint $\tilde{X}_t^{[k]} \neq \emptyset$ and the resulting normalization constant η . Without this constraint, the algorithm could simply make an independent inclusion decision for each singleton with probability equal to $pl[x_{t,i}^{[k]}](z_t)$ (the inclusion decision is made in line 8).

However, because of the requirement $\tilde{X}_t^{[k]} \neq \emptyset$, the algorithm must avoid a situation in which all inclusion decisions are negative, which could easily happen if all likelihoods are close to 0. Thus, simply repeating the sampling process until the requirement is satisfied could be very inefficient. Instead, the inclusion probability is normalized with a constant η_i if $\tilde{X}_t^{[k]} = \emptyset$ for the i -th singleton (line 4) where η_i is defined as:

$$\eta_i^{-1} = 1 - \prod_{j=i}^{|\hat{X}_t^{[k]}|} (1 - pl[x_{t,j}^{[k]}](z_t)) \text{ if } \tilde{X}_t^{[k]} = \emptyset. \quad (3.25)$$

This corresponds to the normalization constant of the generalized Bayesian theorem defined in Eq. (2.74). Because the set $\{x_{t,j}^{[k]} | j < i\}$ of singletons is not included in the final sample $\tilde{X}_t^{[k]}$ if $\tilde{X}_t^{[k]} = \emptyset$ at the i -th iteration, one is in fact not sampling from $m_{\Theta_t}[z_t, \hat{X}_t^{[k]}]$ but from the restricted distribution $m_{\Theta_t}[z_t, \{x_{t,j}^{[k]} | j \geq i\}]$. This is why, for the normalization constant η_i , only the singletons $\{x_{t,j}^{[k]} | j \geq i\}$ are considered. Note that the right-hand side in Eq. (3.25) never becomes 0 because the importance weight $pl[z_t](\hat{X}_t^{[k]})$ is assumed to be always positive, which means that at least one singleton has a positive likelihood.

A way of visualizing this algorithm is the binary decision tree depicted in Fig. 3.7. Each node represents a possible assignment to $\tilde{X}_t^{[k]}$ for the corresponding iteration. The leaf nodes represent the possible final outcomes. Because the leaf node $\tilde{X}_t^{[k]} = \emptyset$ is invalid, its associated mass $\prod_{x_{t,i} \in \hat{X}_t^{[k]}} (1 - pl[x_{t,i}](z_t))$ must be subtracted from the probability of entering a sub-tree containing this node by normalizing with the total mass of the remaining leaf nodes.

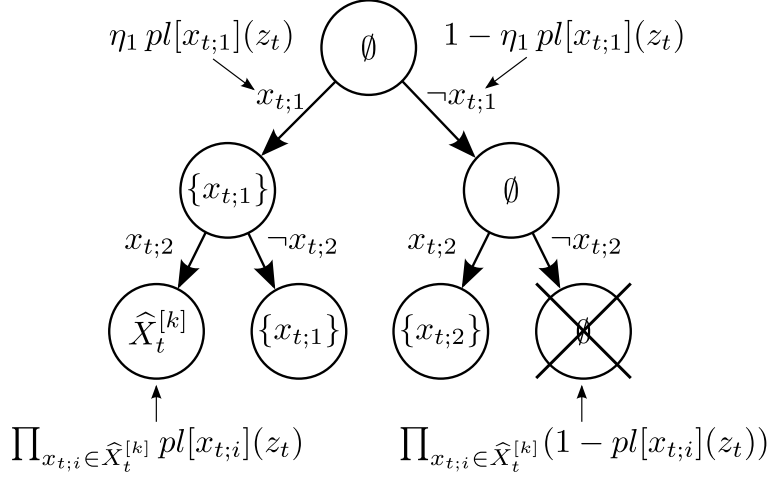


Figure 3.7.: Illustration of sampling from $m_{\Theta_t}[z_t, \hat{X}_t^{[k]}]$. Assuming that $\hat{X}_t^{[k]} = \{x_{t,1}, x_{t,2}\}$, there are 4 possible outcomes, of which $\hat{X}_t^{[k]} = \emptyset$ is invalid. This corresponds to two subsequent binary decisions of including $x_{t,1}$ and $x_{t,2}$ into the sample where $\eta_1 pl[x_{t,1}](z_t)$ represents the inclusion probability for $x_{t,1}$ and $1 - \eta_1 pl[x_{t,1}](z_t)$ the corresponding exclusion probability. The expressions $\prod_{x_{t,i} \in \hat{X}_t^{[k]}} pl[x_{t,i}](z_t)$ and $\prod_{x_{t,i} \in \hat{X}_t^{[k]}} (1 - pl[x_{t,i}](z_t))$ are the probabilities (without normalization) for the two outcomes $\tilde{X}_t^{[k]} = \hat{X}_t^{[k]}$ and $\tilde{X}_t^{[k]} = \emptyset$ respectively (the latter outcome is impossible due to normalization).

3.4.4. Derivation

Here, the algorithms presented above are derived in a more formal manner. For each algorithm, it is shown that the generated samples are in fact drawn from the correct distribution at time t assuming that the same applies to the samples at time $t - 1$.

Prediction Step

Assume that the assignment probability for the prior sample $X_{t-1}^{[k]}$ is given by the true prior distribution with

$$P(X_{t-1}^{[k]} \leftarrow X_{t-1} | z_{0:t-1}) = m[z_{0:t-1}](X_{t-1}), \quad \forall X_{t-1} \subseteq \Theta_{t-1}. \quad (3.26)$$

This means that random variable $X_{t-1}^{[k]}$ is distributed according to the probability distribution over the power set of the state space described by $m_{\Theta_{t-1}}[z_{0:t-1}]$.

Given an assignment $X_{t-1}^{[k]} \leftarrow X_{t-1}$ to the prior sample, the algorithm shown in Fig. 3.2 generates a new proposal sample $\hat{X}_t^{[k]}$ by independently drawing samples $\hat{X}_{t,i}^{[k]}$ according to the assignment probability defined by Eq. (3.19) for

all singletons $x_{t-1;i} \in X_{t-1}$ of the prior sample (line 5). The union of these samples $\hat{X}_{t;i}^{[k]}$ forms the proposal sample $\hat{X}_t^{[k]} = \bigcup_i \hat{X}_{t;i}^{[k]}$ (line 6). Therefore, the assignment probability $P(\hat{X}_t^{[k]} \leftarrow \hat{X}_t | X_{t-1}^{[k]} \leftarrow X_{t-1})$ is given by summing over all possible unions equal to \hat{X}_t where each involved set results from an independent sampling process (the probabilities can thus simply be multiplied). The equality to the transition model mass $m_{\Theta_t}[X_{t-1}]$ then follows from the singleton-conditioned assignment probability defined by Eq. (3.19) and the definition of the disjunctive rule of combination in Eq. (2.60).

$$P(\hat{X}_t^{[k]} \leftarrow \hat{X}_t | X_{t-1}^{[k]} \leftarrow X_{t-1}) = \sum_{\bigcup_{i: x_{t-1;i} \in X_{t-1}} \hat{X}_{t;i} = \hat{X}_t} \prod_{x_{t-1;i} \in X_{t-1}} P(\hat{X}_{t;i}^{[k]} \leftarrow \hat{X}_{t;i} | x_{t-1;i}^{[k]} \leftarrow x_{t-1;i}) \quad (3.27)$$

$$\stackrel{(3.19)}{=} \sum_{\bigcup_{i: x_{t-1;i} \in X_{t-1}} \hat{X}_{t;i} = \hat{X}_t} \prod_{x_{t-1;i} \in X_{t-1}} m[x_{t-1;i}](\hat{X}_{t;i}) \quad (3.28)$$

$$\stackrel{(2.60)}{=} m[X_{t-1}](\hat{X}_t) \quad (3.29)$$

The assignment probability for the proposal sample $\hat{X}_t^{[k]}$ given past observations $z_{0:t-1}$ can be obtained via conditioning by possible assignments to the prior sample (in addition to exploiting conditional independence resulting from the Markov assumption).

$$P(\hat{X}_t^{[k]} \leftarrow \hat{X}_t | z_{0:t-1}) = \sum_{X_{t-1} \subseteq \Theta_{t-1}} P(X_{t-1}^{[k]} \leftarrow X_{t-1} | z_{0:t-1}) P(\hat{X}_t^{[k]} \leftarrow \hat{X}_t | X_{t-1}^{[k]} \leftarrow X_{t-1}) \quad (3.30)$$

By plugging Eq. (3.26) and Eq. (3.29) into Eq. (3.30), one obtains the following equation, which states that the k -th proposal sample is indeed distributed according to the true proposal distribution when assuming that the prior sample is distributed according to the true prior distribution.

$$P(\hat{X}_t^{[k]} \leftarrow \hat{X}_t | z_{0:t-1}) = m[z_{0:t-1}](\hat{X}_t), \quad \forall \hat{X}_t \subseteq \Theta_t \quad (3.31)$$

Correction Step

Assume that the proposal sample $\hat{X}_t^{[k]}$ is distributed according to the true proposal distribution as shown in Eq. (3.31). Furthermore, assume that the compatible sample $\tilde{X}_t^{[k]}$ is distributed according to the observation-induced distribution $m[z_t, \hat{X}_t^{[k]}]$ conditioned by the corresponding proposal sample (shown below):

$$P(\tilde{X}_t^{[k]} \leftarrow \tilde{X}_t | z_t, \hat{X}_t^{[k]} \leftarrow \hat{X}_t) = m[z_t, \hat{X}_t](\tilde{X}_t), \quad \forall \tilde{X}_t \subseteq \hat{X}_t. \quad (3.32)$$

The assignment probability $P(X_t^{[k]} \leftarrow X_t | z_{0:t})$ for a sample $X_t^{[k]}$ in the correction step algorithm shown in Fig. 3.4 is given by summing over all possible assignments to the proposal sample $\hat{X}_t^{[k]}$ and exploiting the Markov assumption (see Eq. (3.33)). The assignment probability $P(X_t^{[k]} \leftarrow X_t | z_t, \hat{X}_t^{[k]} \leftarrow \hat{X}_t)$ conditioned on a particular value \hat{X}_t results from generating a compatible sample $\tilde{X}_t^{[k]}$ (line 4) and performing a resampling step (lines 8–11) with probability proportional to the importance weight $pl[z_t](\hat{X}_t)$ (see Eq. (3.34)).

$$\begin{aligned} & P(X_t^{[k]} \leftarrow X_t | z_{0:t}) \\ &= \sum_{\hat{X}_t \subseteq \Theta_t} P(X_t^{[k]} \leftarrow X_t | z_t, \hat{X}_t^{[k]} \leftarrow \hat{X}_t) P(\hat{X}_t^{[k]} \leftarrow \hat{X}_t | z_{0:t-1}) \end{aligned} \quad (3.33)$$

$$= \eta \sum_{\hat{X}_t \subseteq \Theta_t} P(\tilde{X}_t^{[k]} \leftarrow X_t | z_t, \hat{X}_t^{[k]} \leftarrow \hat{X}_t) pl[z_t](\hat{X}_t) P(\hat{X}_t^{[k]} \leftarrow \hat{X}_t | z_{0:t-1}) \quad (3.34)$$

(Normalization is required because the resampling is performed with probability *proportional* to the importance weights.)

Using the two assumptions defined by Eq. (3.31) and Eq. (3.32) regarding the distribution of the proposal sample and the compatible sample results in the following expression:

$$= \eta \sum_{\hat{X}_t \subseteq \Theta_t} m[z_t, \hat{X}_t](X_t) pl[z_t](\hat{X}_t) m[z_{0:t-1}](\hat{X}_t). \quad (3.35)$$

Finally, by applying the f -total law defined by Eq. (2.63) and Eq. (2.64), one obtains the equality which states that sample $X_t^{[k]}$ is distributed according to the true mass distribution resulting from the correction step defined by Eq. (3.10).

$$= m[z_{0:t}](X_t^{[k]}) \quad (3.36)$$

This last step also highlights the importance of the f -total law as it forms the basis for combining mass functions using importance sampling.

Generating Compatible Samples

The function `compatible_sample` shown in Fig. 3.6 distinguishes between two cases when deciding whether to include singleton $x_{t,i}^{[k]} \in \hat{X}_t^{[k]}$ into random sample $\tilde{X}_t^{[k]}$. Let n denote the cardinality of the proposal sample with $n = |\hat{X}_t^{[k]}|$ and let $\tilde{X}_{t,i}^{[k]}$ denote the partially constructed sample $\tilde{X}_t^{[k]}$ right before entering the loop in line 2 for the i -th iteration (starting with $i = 1$). Depending on whether $\tilde{X}_{t,i}^{[k]} = \emptyset$, the inclusion probability for singleton $x_{t,i}^{[k]}$ is either given by

the normalized or by the unnormalized likelihood $pl[x_{t,i}^{[k]}](z_t)$ (lines 3–10).

$$P(x_{t,i}^{[k]} \in \tilde{X}_t^{[k]} | z_t, \tilde{X}_{t,i}^{[k]} \neq \emptyset) = pl[x_{t,i}^{[k]}](z_t) \quad (3.37)$$

$$P(x_{t,i}^{[k]} \in \tilde{X}_t^{[k]} | z_t, \tilde{X}_{t,i}^{[k]} = \emptyset) = \eta_i pl[x_{t,i}^{[k]}](z_t) \quad (3.38)$$

$$\eta_i^{-1} = 1 - \prod_{j=i}^n (1 - pl[x_{t,j}^{[k]}](z_t)) \quad (3.39)$$

Let c be the index of the first singleton that is included in the final sample $\tilde{X}_t^{[k]}$. Furthermore, let $\Theta_{t,c}$ denote the remaining frame of discernment given c , i.e., the frame of discernment after having learned that the first $c - 1$ singletons are not included in $\tilde{X}_t^{[k]}$.

$$c = \min\{i | x_{t,i}^{[k]} \in \tilde{X}_t^{[k]}, 1 \leq i \leq n\} \quad (3.40)$$

$$\Theta_{t,c} = \{x_{t,i} | x_{t,i} \in \hat{X}_t^{[k]}, i \geq c\} \quad (3.41)$$

Note that c exists because even if $\tilde{X}_{t,n}^{[k]} = \emptyset$, the final singleton is guaranteed to be included because of $\eta_n^{-1} = pl[x_{t,n}^{[k]}](z_t)$, thus satisfying the constraint $\tilde{X}_t^{[k]} \neq \emptyset$ (assuming $pl[x_{t,n}^{[k]}](z_t) > 0$).

$$P(x_{t;n}^{[k]} \in \tilde{X}_t^{[k]} | \tilde{X}_{t;n}^{[k]} = \emptyset) = \eta_n pl[x_{t;n}^{[k]}](z_t) = 1 \quad (3.42)$$

For any given value c , it can then be shown that the final sample $\tilde{X}_t^{[k]}$ is distributed according to the mass function $m_{\Theta_t}[z_t, \Theta_{t,c}]$. With c known, the first $c - 1$ singletons can be ignored because they are not part of the final sample. For the c -th singleton, the likelihood is normalized by η_c according to Eq. (3.38) and Eq. (3.39). All subsequent inclusion probabilities are unnormalized according to Eq. (3.37). As a result, the sample assignment probability $P(\tilde{X}_t^{[k]} \leftarrow \tilde{X}_t | z_t, c)$ can be expressed in terms of the singleton inclusion/exclusion probabilities and the equality to the mass function $m[z_t, \Theta_{t,c}](\tilde{X}_t)$ defined by Eq. (3.23) follows.

$$\begin{aligned} & P(\tilde{X}_t^{[k]} \leftarrow \tilde{X}_t | z_t, c) \\ &= \eta_c \prod_{x_{t,i} \in \tilde{X}_t} pl[x_{t,i}^{[k]}](z_t) \prod_{x_{t,i} \in \tilde{\tilde{X}}_t} (1 - pl[x_{t,i}^{[k]}](z_t)) \end{aligned} \quad (3.43)$$

$$= m[z_t, \Theta_{t,c}](\tilde{X}_t), \quad \forall \tilde{X}_t \subseteq \Theta_{t,c} \quad (3.44)$$

3.5. Complexity and Approximation Error

The computational complexity of performing exact inference for the prediction and correction step is exponential (in the worst case) with $O(2^{|\Theta|})$, regarding both time and space.⁴ This is because, in the prediction step in Eq. (3.7) as

⁴The time index of the state space is omitted here because the state space is assumed to be time-invariant.

well as in the correction step in Eq. (3.10), Dempster's rule of combination has to be computed over the frame of discernment Θ , producing up to $2^{|\Theta|} - 1$ focal sets. For the prediction step, the computational effort is actually even a little higher because the transition model has to be constructed from the singleton-conditioned distributions $m_{\Theta_t}[x_{t-1}]$ in Eq. (3.3), thus adding an additional factor $|\Theta|$ for the number of singletons (though this construction could be performed offline).

In contrast, the Monte-Carlo approach is able to compute both steps with time and space complexity $O(K|\Theta|)$. The prediction step algorithm defined in Fig. 3.2 iterates over all particles (with $O(K)$), draws a sample from $m_{\Theta_t}[x_{t-1}]$ for each singleton x_{t-1} , and computes the union of these samples. Implemented naively, the combined complexity of sampling and uniting could be quadratic in $|\Theta|$ (in the worst case). However, singletons that have already been sampled can be ignored in subsequent sampling steps, reducing the complexity to linear because the final proposal sample only has up to $|\Theta|$ elements.

The correction step algorithm shown in Fig. 3.4 iterates twice over the set of particles. In the first loop, the function `compatible_sample` is called, which has a complexity of $O(|\Theta|)$ because it iterates over all singletons of a sample. In the second loop, each resampling iteration has complexity $O(1)$ because drawing a sample and adding it to the final sample set only requires a constant amount of time. While a complexity of $O(K|\Theta|)$ for the prediction and correction step is still computationally expensive for large state spaces and high sample counts, it constitutes a very significant improvement over the exponential complexity associated with exact inference and it allows one to solve problems which would otherwise be intractable.

In order to analyze the computational complexity and the approximation error of the Monte-Carlo approach empirically, an experiment is conducted where the computation time and the approximation error is measured for different state space sizes and sample counts. Note that the following setup is completely artificial and only serves to analyze the performance of the algorithms. For the transition model, the state is assumed to remain unchanged between two consecutive time steps with a probability of at least P_{\min} (here, $P_{\min} = 0.6$ is used). With respect to the remaining mass, one is entirely ignorant and, as a consequence, mass $1 - P_{\min}$ is assigned to the state space Θ . For simplicity, the observation space is defined to be equal to the state space. The observation model assigns a mass of P_{\min} to the current state while the remaining mass is assigned to the frame of discernment. As a result, the transition and observation models can be described by the following equations:

$$m[x_{t-1}](X_t = \{x_{t-1}\}) = m[x_t](Z_t = \{x_t\}) = P_{\min}, \quad (3.45)$$

$$m[x_{t-1}](X_t = \Theta) = m[x_t](Z_t = \Theta) = 1 - P_{\min}. \quad (3.46)$$

Three evidential filter algorithms are compared in the following: exact evidential filtering, evidential particle filtering using 100 samples, and evidential

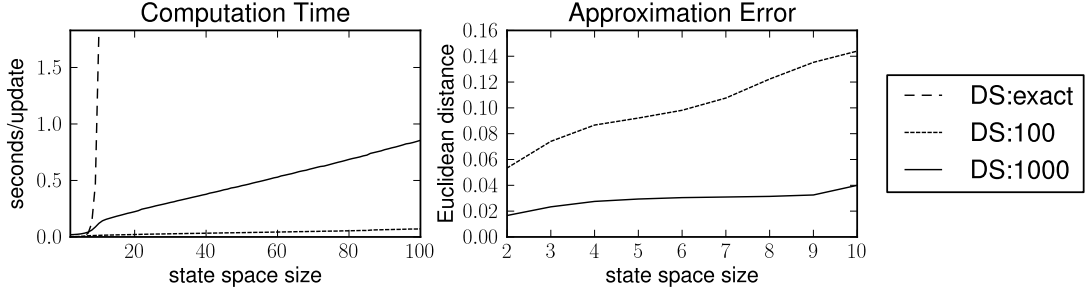


Figure 3.8.: Mean computation time required for 10 update iterations (left) and mean approximation error with respect to the true distribution (right) for different state space sizes and sample counts. **DS:exact** denotes the performance of exact evidential inference while **DS:100** and **DS:1000** denote the evidential particle filtering performance with 100 and 1000 samples. (Figure reprinted from [Reineking, 2011].)

particle filtering using 1000 samples. Different state space sizes are used ranging from 2 up to 100. For each algorithm and state space size, 10 update iterations are performed where one update consists of a prediction step followed by a correction step. The underlying observations $z_{0:t}$ and true states $x_{0:t}$ are sampled from the pignistic transformations of the models defined in Eq. (3.45) and (3.46) (the initial state is chosen randomly and the initial prior for each algorithm is vacuous). This process is repeated 100 times and the results are averaged.

Fig. 3.8 shows the computation time required for performing 10 updates using evidential particle filtering compared to the time associated with exact inference for different state space sizes. As seen, the computation time of exact inference grows exponentially and, as a consequence, it is only measured for state space sizes up to 10. In contrast, the update times of the particle filter algorithms grow proportionally with respect to the state space size. (The non-linear increase for smaller state spaces is a result of the implementation handling duplicate samples more efficiently.) For small state space sizes (up to 8), exact inference is faster than particle filtering (using 1000 samples) because of the high factor K in the particle filter complexity $O(K|\Theta|)$.

In addition to the computation time, the figure also shows the approximation error with respect to the exact solution. The error is measured as the Euclidean distance between two mass functions where each mass function is represented as a vector (each non-empty subset of the state space represents a component in the vector, i.e., the vector has $2^{|\Theta|} - 1$ components). Because exact inference is only performed for state space sizes up to 10, the approximation error is only measured up to this point. As expected, the approximation error grows with increasing state space size, in particular because the true mass functions are not very sparse. However, the absolute error is still quite small, especially for the

approximation based on 1000 samples.

3.6. Application to Bearings-only Tracking

In this section, the evidential particle filter algorithm is applied to a simulation-based 2D tracking problem and compared to a Bayesian solution. Measurements from a single sensor are assumed to only provide information about the direction of the target object, resulting in a bearings-only tracking problem [La Scala and Morelande, 2008].⁵ Bearings-only information usually leads to high uncertainty and is difficult to capture using probabilistic particle filters because the particle set has to cover large parts of the state space. In contrast, a high degree of uncertainty (if it corresponds to ignorance rather than conflict) can be expressed very well using an evidential particle filter where each particle can represent an arbitrary subset of the state space.

In order to estimate the position of a target object using only a bearing sensor, a model of the target object’s motion is required. In many bearings-only tracking scenarios, the target behavior is highly constrained due to momentum and limited maneuverability (e.g., for tracking of airplanes or ships) and the underlying physics along with Gaussian noise assumptions can be used to devise accurate motion models [Bar-Shalom et al., 2004]. Here, the target behavior is assumed to be governed by a more “high-level” process where the target object only has a vague destination in which it moves (e.g., a person walking in a certain direction). The possible destinations in this example are the cardinal directions *north*, *west*, *south*, and *east*. For simplicity, the target’s destination is assumed to remain unchanged over time. The advantage of belief functions in this scenario is that they allow the observer to remain ignorant about unknown parameters of the underlying models.

Because the destination is unknown to the observer, it has to be estimated, resulting in a joint estimation of destination and position (positions are expressed in Cartesian coordinates with origin at the sensor position). A possible solution to such a joint tracking problem is a set of destination-matched filters where one filter algorithm is used for each destination [Gordon et al., 2002, Ristic et al., 2004]. However, such an approach exhibits certain limitations and it is not considered here (e.g., destination changes, though not considered here, cannot be modeled). In order to model the tracking process using belief functions, the position is discretized using a 100×100 grid, resulting in a joint state space size of 40000.

State transitions are modeled as follows. Let $x_t = (x_{t;x}, x_{t;y}, x_{t;d})$ denote the target object’s current state where $(x_{t;x}, x_{t;y}) \in \mathbb{Z}^2$ represents the discrete position and $x_{t;d} \in \{n, w, s, e\}$ represents the destination (i.e., a cardinal direction).

⁵In [Schult et al., 2013], a bearings-only tracking approach based on a probabilistic particle filter is proposed for audio-visual source localization. An application of evidential filtering to vision-based self-localization is presented in [Reineking et al., 2010].

It is assumed that the target object usually moves in the general direction of its destination. If a movement would cause the object to move out of the grid, the state is assumed to remain unchanged. Let $A(x_t)$ denote the area where the object will likely move to next given the current state x_t .

$$A(x_t) = \begin{cases} \{(x', x_{t;y} - 1) | x' \in \mathbb{Z} \wedge |x' - x_{t;x}| \leq 1\} & \text{if } x_{t;d} = n, \\ \{(x', x_{t;y} + 1) | x' \in \mathbb{Z} \wedge |x' - x_{t;x}| \leq 1\} & \text{if } x_{t;d} = s, \\ \{(x_{t;x} - 1, y') | y' \in \mathbb{Z} \wedge |y' - x_{t;y}| \leq 1\} & \text{if } x_{t;d} = w, \\ \{(x_{t;x} + 1, y') | y' \in \mathbb{Z} \wedge |y' - x_{t;y}| \leq 1\} & \text{if } x_{t;d} = e \end{cases} \quad (3.47)$$

In addition, the object may “pause” at any time with some probability, in which case the object remains in the same state. Let $R \subseteq \Theta_R$ with $\Theta_R = \{0, 1\}$ denote whether the object remains in the same state ($R = 1$) or not ($R = 0$). The probability $P(R = 1) = \pi_R$ is assumed to be unknown.⁶ The resulting evidential transition model reflects the ignorance regarding R by assuming a vacuous distribution over it with $m(\Theta_R) = 1$. In contrast, a corresponding probabilistic transition model has to commit to a value for π_R , in this case, a uniform distribution is assumed.⁷ In addition, the distribution over cells in $A(x_t)$ has to be assumed to be uniform. The probabilistic transition model is obtained by conditioning on R .

$$P(x_t | x_{t-1}) = \sum_{r \in \Theta_R} P(x_t | x_{t-1}, r) P(r) \quad (3.48)$$

$$= \begin{cases} \frac{1}{2|A(x_t)|} & \text{if } (x_{t;x}, x_{t;y}) \in A(x_{t-1}), \\ \frac{1}{2} & \text{if } (x_{t;x}, x_{t;y}) = (x_{t-1;x}, x_{t-1;y}), \\ 0 & \text{else} \end{cases} \quad (3.49)$$

Similarly, the evidential transition model results from conditioning on R and applying the disjunctive rule of combination for the case $R = \Theta_R$.

$$m[x_{t-1}](X_t) \stackrel{(2.63)}{=} \sum_{R \subseteq \Theta_R} m[x_{t-1}, R](X_t) m(R) \quad (3.50)$$

$$m[x_{t-1}](X_t) \stackrel{(2.60)}{=} (A(x_{t-1}) \times \{x_{t-1;d}\}) \cup \{x_{t-1}\} = 1 \quad (3.51)$$

The result of conditioning on R is that the probabilistic model is biased towards remaining in the same state (with mass 0.5) while the evidential model remains agnostic regarding both possibilities.

For the bearing measurements, a Gaussian-like noise model is assumed. The expected value is equal to the target object’s angle with respect to the sensor

⁶Indeed, state transition probabilities are difficult to estimate for many practical problems such as activity recognition [Duong et al., 2005].

⁷Modeling π_R as a random variable would be an alternative but this creates new problems as it increases the number of variables.

position (s_x, s_y) and the standard deviation is $\sigma = 0.15$. In order to cope with plausibilities of continuous measurements z_t , the following distribution from [Aregui and Denc  ux, 2008] is used instead of an ordinary Gaussian (it corresponds to the “commonality-least-committed” belief function whose pignistic transformation is a Gaussian with the specified parameters, see also Sect. 5.4.5):

$$pl[x_t](z_t; \mu_t, \sigma) = \begin{cases} \frac{2(z_t - \mu_t)}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(z_t - \mu_t)^2}{2\sigma^2}\right) + 2(1 - \Phi(\frac{z_t - \mu_t}{\sigma})) & \text{if } z_t \geq \mu_t, \\ \frac{2(\mu_t - z_t)}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(z_t - \mu_t)^2}{2\sigma^2}\right) + 2\Phi(\frac{z_t - \mu_t}{\sigma}) & \text{else,} \end{cases} \quad (3.52)$$

$$\mu_t = \text{atan2}(x_{t;y} - s_y, x_{t;x} - s_x). \quad (3.53)$$

Here, Φ is the standard normal cumulative distribution and atan2 (a quadrant-sensitive version of the arctan function) computes the angle of the target with respect to the sensor. For simplicity, the same sensor model is used for evidential and Bayesian filtering. The fact that Eq. (3.52) is technically not a probability density distribution (the integral is generally not 1) does not matter here because the Bayesian filter algorithm is invariant with respect to scaling of likelihoods if normalization is performed afterwards.

For the simulation, the true state sequence and corresponding measurements are sampled from the models defined by Eq. (3.48) and (3.52). For transitions, the true probability distribution over R is assumed to be significantly different from a uniform one with $\pi_R = 0.1$. The evidential particle filter is compared to a Bayesian particle filter and to exact Bayesian filtering. All filter algorithms process the same measurements. The initial belief is vacuous for the evidential particle filter and uniform for the Bayesian filters. The sample count for the evidential particle filter is set to $K = 100$ while, for the Bayesian particle filter, two runs are performed, one using $K = 100$ and the other using $K = 1000$.

Fig. 3.9 shows the estimated marginal position plausibility $pl[z_{0:t}](x_{t;x}, x_{t;y})$ resulting from the evidential particle filter at different points in time. In Fig 3.9a, the plausibility after the first measurement is shown. Here, the observer is entirely ignorant regarding the distance of the object and can only determine a range of plausible angles due to measurement noise. The angular uncertainty decreases with additional measurements (Fig 3.9b and 3.9c) until the object is very close to the sensor and the position can be uniquely determined (Fig 3.9d). In Fig. 3.9c, it can also be seen that ignorance decreases and the distribution becomes more “Bayesian” with sufficiently many measurements. Afterwards, the uncertainty (including ignorance) increases again (Fig 3.9e and 3.9f), although the uncertainty is lower than in the beginning because, at this point, the true destination “east” has been uniquely determined.

The process of successively ruling out possible target destinations is shown in Fig. 3.10 where the marginal destination plausibility $pl[z_{0:t}](x_{t;d})$ is plotted over time. Like for positions, the initial belief is vacuous. First, “north” is ruled out because the target object moves slightly in a southern direction, which the sensor can directly measure from its position. The same applies to “south”

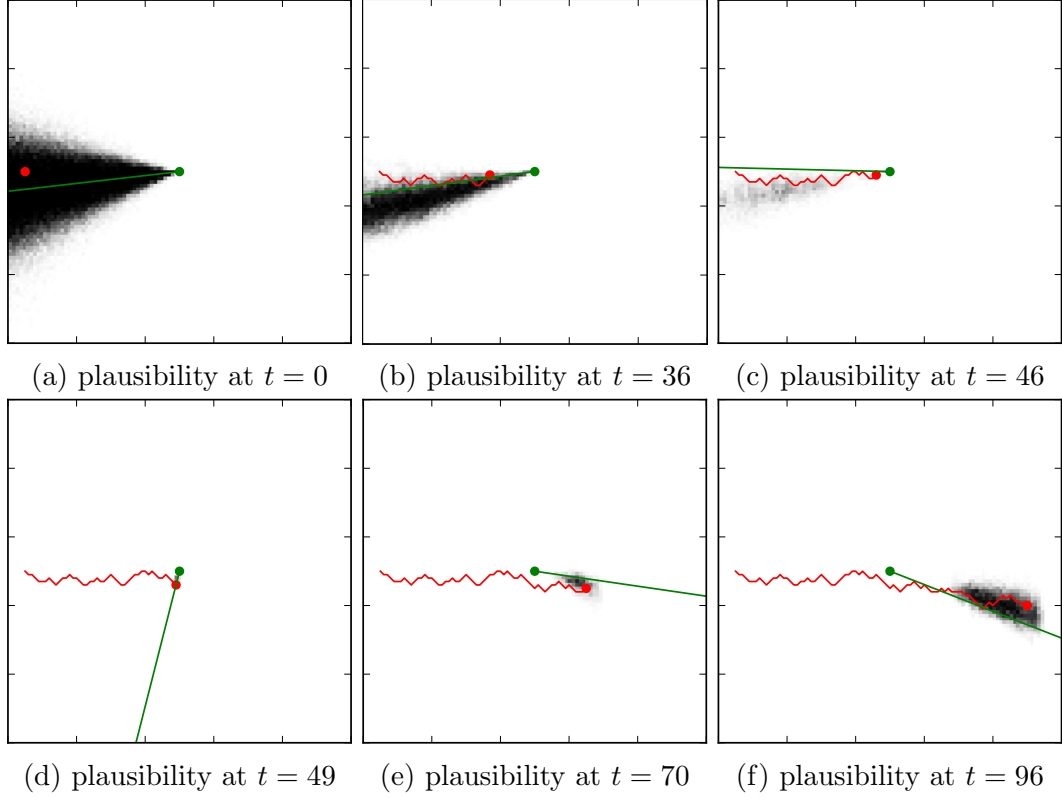


Figure 3.9.: Marginal position plausibility $pl[z_{0:t}](x_{t;x}, x_{t;y})$ of each grid cell at different points in time resulting from the evidential particle filter. Higher grid cell plausibilities are visualized as darker colors. The green point at the center shows the sensor position and the green line represents a bearing measurement. The red point is the target object and the red line is its path.

(though not immediately) while destination “west” is ruled out once the object is very close to the sensor and the eastward direction can be measured.

In order to compare the results of the evidential approach to the Bayesian solutions, the tracking error is plotted over time in Fig. 3.11. The error for each point in time t is defined as the expected Euclidean distance $E(\|T_t - X_t\|)$ between the true target position T_t and random variable X_t which is distributed according to the marginal position probability distribution (obtained via the pignistic transformation of the marginal position distribution for the evidential particle filter). The probabilistic particle filter with $K = 100$ is unable to cope with the high initial uncertainty because no sample is close enough to the true state and the corresponding error grows steadily over time. For the other filter algorithms, the initial decrease of the error until around $t = 15$ is caused by multiple measurements reducing the angular uncertainty. The subsequent increase until around $t = 48$ is the result of the object moving closer to the sensor without the sensor being able to determine its distance. Once the

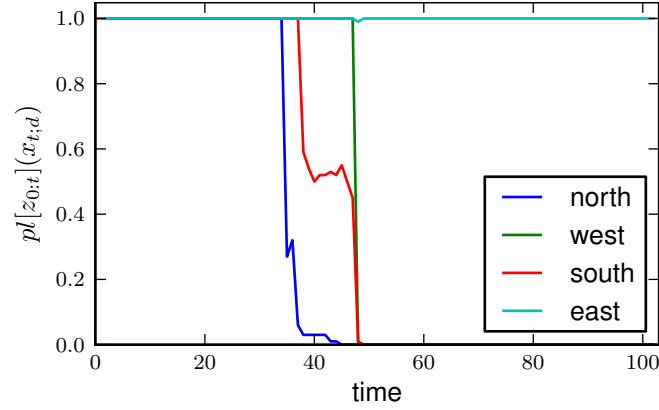


Figure 3.10.: Marginal destination plausibility over time. The initial destination belief is vacuous but, over time, all destinations except for “east” are ruled out.

distance has been determined, the error becomes very small for all approaches (except for the probabilistic particle filter with $K = 100$). After this, the error develops differently: the evidential particle filter exhibits a smaller increase in error while the error of the Bayesian filters increases more quickly. This is caused by the Bayesian transition models where the assumption of a uniform distribution for R results in a bias for remaining in the same state while the true change probability is higher.

Finally, Fig. 3.12 shows the expected tracking error averaged over time for different values of the true transition parameter π_R . In addition to averaging over time, for each value of π_R and each algorithm, the filtering process is repeated five times and the results are averaged as well. The probabilistic particle filters are not shown because they are only approximations of exact Bayesian filtering and their results are generally worse and not very stable as they are not always able to track the object (even for $K = 1000$). As expected, the exact Bayesian solution outperforms the evidential one if the true value of π_R is close to the assumed value of 0.5. However, considering the entire range of possible values for π_R , the evidential particle filter outperforms the Bayesian solution in terms of the expected tracking error, showing that explicitly modeling ignorance provides an advantage in this scenario.

3.7. Discussion

The evidential filter presented in Sect. 3.3 and the evidential particle filter presented in Sect. 3.4 are both generalizations of their respective discrete Bayesian counterparts. Like most Bayesian particle filter algorithms, the evidential particle filter uses importance sampling to incorporate new observations. While this only requires weighting of particles in the Bayesian case, for belief functions, it

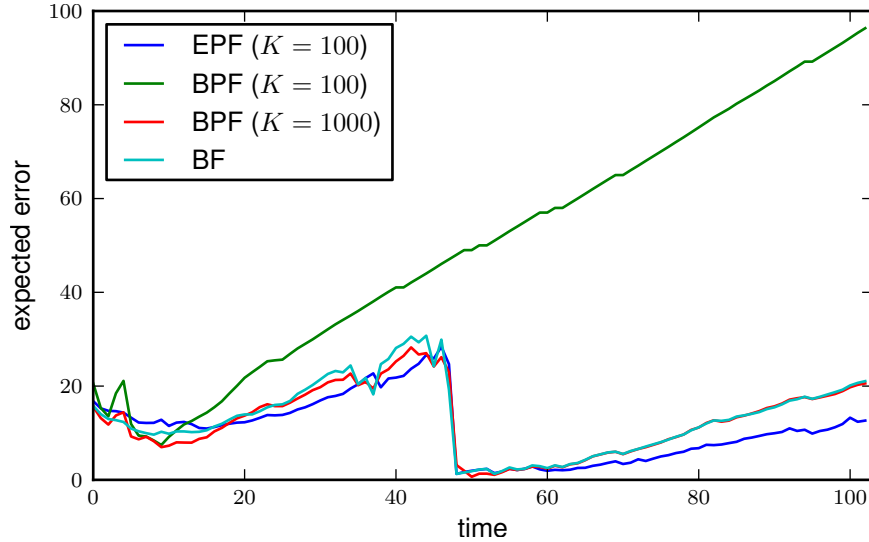


Figure 3.11.: Tracking error (expected Euclidean distance between estimated and true target position) over time. “EPF ($K = 100$)” represents the evidential particle filter, “BPF ($K = 100$)” and “BPF ($K = 1000$)” represent the Bayesian particle filters, and “BF” represents the analytical Bayesian filter solution.

also involves constructing a subset of each focal set, showing the additional set-based dimension of uncertainty associated with belief functions. For a Bayesian filter, the prediction step generally results in an increase in uncertainty while the correction step reduces uncertainty. The same applies to evidential filtering and its two dimensions of uncertainty (conflict and ignorance) where the prediction step generally causes an increase in both dimensions (conflict grows and the focal set cardinalities grow as well due to the disjunctive rule) while the correction step reduces uncertainty in both dimensions (conflict is reduced by weighting and Dempster’s rule causes focal set cardinalities to become smaller).

One problem that is not explicitly addressed in this chapter but that often plays a role in applications of probabilistic particle filters is that of *particle deprivation* [Thrun et al., 2005, chapter 4]. This problem occurs if the true state is not represented by any particle (like in the previous section for the probabilistic particle filter with $K = 100$). While choosing a larger value for K reduces this problem, it also increases the computational costs. A common approach for probabilistic particle filters is therefore to introduce a small number of random samples in the correction step. However, the probability of such a random sample being close or equal to the true state is quite small for large state spaces. In contrast, when using an evidential particle filter, such samples would not have to be drawn randomly but could instead represent the entire state space Θ_t , in which case the true state would be guaranteed to be included.

While the presented evidential particle filter algorithm can be applied to any

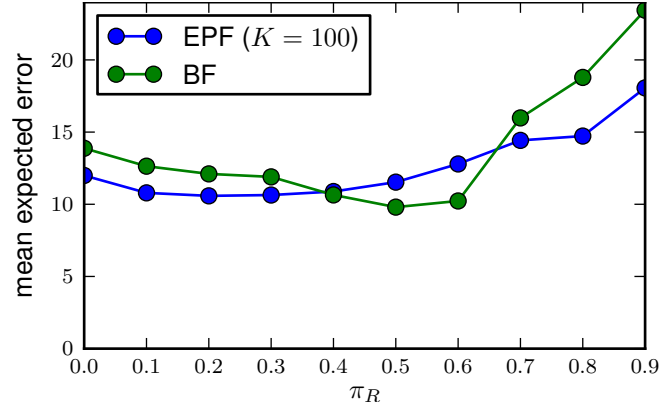


Figure 3.12.: Expected tracking error averaged over time and five runs for different values of π_R . “EPF ($K = 100$)” represents the evidential particle filter while “BF” represents the exact Bayesian filter solution.

state filtering problem with discrete state variables, being able to handle continuous states would be of great practical importance for making the approach more broadly applicable. In contrast, continuous measurements can be handled because the algorithm only requires likelihood values and the belief function over the domain of z_t is never explicitly constructed. In fact, multiple works have been presented over the years which extend belief functions to the domain of real numbers. For example, in [Liu, 1996], Gaussian belief functions are introduced which, in case of filtering, would result in an evidential Kalman filter similar to the one proposed in [Smets and Ristic, 2007] (with all the usual limitations associated with Kalman filters mentioned at the beginning of this chapter). Another approach is based on only considering closed intervals as focal sets [Smets, 2005a]. However, this approach only works for conjunctive combinations of belief functions and not for a disjunctive combination because, in contrast to intersections, the union of two closed intervals does not generally yield a closed interval. For this reason, the disjunctive rule of combination could not be used for constructing the transition model in the prediction step in Eq. (3.3).

This equation also contains an iteration over all elements of the domain, which, in case of real numbers, could only be computed for special cases. The same problem applies to the generalized Bayesian theorem used in the correction step. Thus, in order to apply the evidential particle filtering approach to continuous state variables, these basic tools of belief function theory would have to be extended and it remains an open research question.

4

Evidential SLAM

4.1. Introduction

In this chapter, a belief-function-based solution to the problem of *simultaneous localization and mapping* (SLAM) is presented. The SLAM problem consists of a mobile robot building a spatial representation of its environment (usually just referred to as a “map”) while at the same time localizing itself using this spatial representation. The theoretical basis for the evidential SLAM approach and some of the empirical results shown in this chapter were originally presented in [Reineking and Clemens, 2013].

SLAM is considered to be one of the most fundamental problems in robotics [Durrant-Whyte and Bailey, 2006]. Thus, when the first solutions to the SLAM problem were proposed in [Smith and Cheeseman, 1986, Smith et al., 1990], it marked an important breakthrough in the field. Whenever a mobile robot has to explore a new environment autonomously, it has to solve the SLAM problem in order to navigate reliably in the environment. What makes the SLAM problem difficult is that the two processes of localization and mapping are inherently linked because localization is not possible without a map and mapping is not possible without localization information. As a result, SLAM has to be modeled as a joint estimation problem where the robot’s location and the map are estimated together.

Regarding the spatial representation, there are two common approaches: feature-based maps and grid-based maps [Thrun, 2002]. Feature-based maps consist of discrete *landmarks* which can, for example, be tracked using Kalman filters. In contrast, grid-based maps discretize the environment into cells using a regular grid structure. The most popular type of grid maps are occupancy grid maps where each cell has a probability of being occupied [Elfes, 1989,

Moravec, 1988]. Occupancy grid maps work particularly well in combination with range sensors like sonar. A major advantage of grid maps is that they provide an explicit representation of free space which is essential for navigation.

The state of a grid cell is usually described probabilistically, a single probability in case of an occupancy grid map. In contrast, belief functions can make additional dimensions of uncertainty explicit in the map. A lack of evidence regarding the state of a cell can be distinguished from contradictory sensor measurements (e.g., a vacuous belief function vs. a uniform Bayesian belief function). The amount of conflict resulting from multiple measurements can be made explicit by using unnormalized belief functions. In addition, evidential maps allow for the use of different combination rules. Most importantly though, computational complexity is not an issue because of the small frame of discernment where, in case of an occupancy grid map, each cell can only be in one of two states. For these reasons, there exist a number of works on occupancy grid mapping based on belief functions [Moras et al., 2011, Yang and Aitken, 2006, Mullane et al., 2006, Ribo and Pinz, 2001, Li et al., 2007, Pagac et al., 1998].

However, all the works on evidential mapping have in common that they only tackle the mapping part of SLAM and ignore the localization part. Therefore, an evidential solution to the entire SLAM problem is presented in this chapter. The focus here is on 2D occupancy grid maps, although the approach could be extended to 3D or non-binary cell representations. The basis for the evidential SLAM solution presented here is the FastSLAM algorithm [Montemerlo et al., 2002], a particularly successful SLAM algorithm which utilizes a particle filter for maintaining a set of map and path hypotheses. In case of probabilistic sensor models, the evidential approach reduces to the classical FastSLAM algorithm.

The remainder of this chapter is structured into four sections. In Sect. 4.2, the SLAM problem is defined theoretically and the probabilistic FastSLAM algorithm is introduced. In Sect. 4.3, the equations for the evidential SLAM approach are derived and the resulting algorithm is described. In Sect. 4.4, the models required by the SLAM algorithm, including evidential forward and inverse models for sonar, are presented. Experimental results including maps generated by different combination rules in different environments are presented in Sect. 4.5. In the final section, the proposed approach is discussed and possible extensions are pointed out.

4.2. Simultaneous Localization and Mapping

The goal of SLAM is to compute the joint distribution of the robot's pose x_t (position and orientation) at time t and the map Y given all sensor measurements $z_{0:t}$ and robot controls $u_{1:t}$.

$$p(x_t, Y | z_{0:t}, u_{1:t}) \tag{4.1}$$

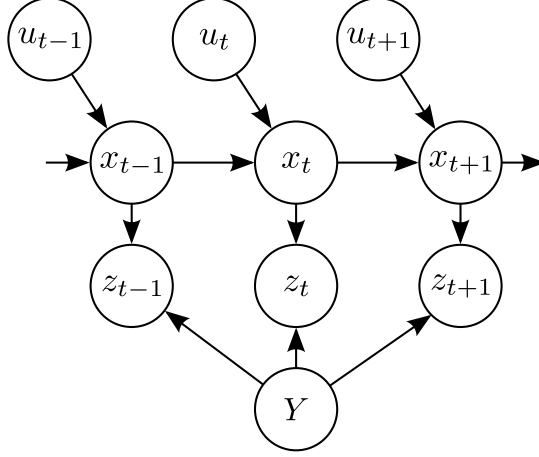


Figure 4.1.: Dynamic belief network for SLAM. The sequence of robot poses $x_{0:t}$ and the map Y cannot be observed directly and have to be estimated based on measurements $z_{0:t}$ and controls $u_{1:t}$.

The upper-case notation for Y is used to indicate that the map is later modeled as an evidential variable. Starting with $t = 0$, there are a total of $t + 1$ poses and measurements over time but only t state transitions and controls, which is why z_0 is the first measurement while u_1 is the first control.

The corresponding belief network is shown in Fig. 4.1. Like in the previous chapter on particle filtering, the poses (states) form a first-order Markov chain with conditionally independent measurements. In addition, the controls provide information about pose changes. The map determines, along with the current pose, what the robot expects to sense. Note that the map is assumed to be time-invariant here, although this could easily be changed.¹

In the 2D case, the pose x_t is a vector composed of three components: a pair of Cartesian coordinates $x_{t;x}, x_{t;y}$ and an angle $x_{t;\phi}$ describing the robot's current orientation.

$$x_t = (x_{t;x}, x_{t;y}, x_{t;\phi})^T \quad (4.2)$$

In case of occupancy grid mapping, a map consists of M binary variables where M denotes the number of grid cells. Let Y_i denote the binary evidential variable corresponding to the i -th cell with frame of discernment $\Theta_Y = \{o, \neg o\}$ where o represents an occupied cell and $\neg o$ represents a free cell. Then the entire map Y is an evidential variable with frame of discernment Θ_Y^M defined as the Cartesian product of all cell spaces.

$$Y_i \subseteq \Theta_Y \text{ with } \Theta_Y = \{o, \neg o\} \quad (4.3)$$

$$Y \subseteq \Theta_Y^M \quad (4.4)$$

¹In case of an evidential map, by applying a version of the evidential prediction step presented in the previous chapter.

For measurements, the focus in this chapter is on range sensors, specifically sonar. A sonar sensor provides a noisy range measurement z_t with respect to the closest obstacle located in the measurement cone. In order to keep notation simple, only a single sensor is considered first, though modeling an array of sonar sensors producing a vector of measurements is a straightforward extension. A range measurement z_t is a real number and sonar generally has a maximum range denoted by z_{\max} .

$$z_t \in \Omega_t \text{ with } \Omega_t = [0, z_{\max}] \quad (4.5)$$

A control u_t describes the robot's motion, i.e., the pose change between time $t - 1$ and t . Here, the control is a two-component vector composed of a velocity $u_{t;v}$ in the direction of the current bearing and an angular velocity $u_{t;w}$.

$$u_t = (u_{t;v}, u_{t;w})^T \quad (4.6)$$

Like range measurements, controls are assumed to be noisy. The noise can either result from noisy odometry measurements (e.g., measuring wheel rotations) or from imperfect execution of an action. From a theoretical standpoint, the origin of the control noise does not make much of a difference and it is sufficient to assume that state transitions are generally noisy.

In order to compute the joint distribution in Eq. (4.1), there are different approaches. Perhaps the most straightforward one is using an extended Kalman filter where the state consists of a pose and a feature-based map. The problem of this approach is that the number of parameters that have to be estimated grows quadratically with the number of landmarks due to the corresponding covariance matrix, which is why it does not scale well. There are multiple other approaches [Thrun et al., 2005] but the focus here is on FastSLAM because it is one of the most popular state-of-the-art SLAM algorithms and because it is the basis for the evidential SLAM algorithm presented in Sect. 4.3.

4.2.1. FastSLAM

The FastSLAM algorithm was originally introduced in [Montemerlo et al., 2002]. For FastSLAM, not only the current pose x_t is estimated but rather the entire path $x_{0:t}$. Estimating the entire path corresponds to the so-called *full* SLAM problem. In contrast, estimating only the current pose as in Eq. (4.1) is called the *online* SLAM problem. In fact, FastSLAM is the only SLAM algorithm capable of solving both the full and the online SLAM problem at the same time [Thrun et al., 2005, chapter 13].

The FastSLAM algorithm is based on a technique called *Rao-Blackwellization* [Doucet et al., 2000] where the joint estimation problem is factorized into a path estimation problem and mapping problem that is conditioned on a specific path. The factorization simply follows from applying the product rule to the joint distribution defined by Eq. (4.1) (using the entire path instead of just the current

pose). The important part is that, given a path, the features/cells in the map become conditionally independent, which greatly simplifies the mapping problem because the map can be represented as a product of marginal distributions.

$$p(x_{0:t}, Y | z_{0:t}, u_{1:t}) = p(x_{0:t} | z_{0:t}, u_{1:t}) p(Y | x_{0:t}, z_{0:t}) \quad (4.7)$$

$$= p(x_{0:t} | z_{0:t}, u_{1:t}) \prod_{i=1}^M p(Y_i | x_{0:t}, z_{0:t}) \quad (4.8)$$

The conditional independence is exact regarding the pose uncertainty with respect to the robot; it is only an approximation if features/cells in the map are correlated regardless of the robot's pose uncertainty (which is usually the case). In addition, the map is conditionally independent of the controls if the path is given.

The joint estimation problem is solved using a particle filter where the conditional mapping problems $p(Y_i | x_{0:t}, z_{0:t})$ are solved analytically. Each particle consists of a path hypothesis $x_{0:t}^{[k]}$ and a corresponding map distribution $p(Y | x_{0:t}^{[k]}, z_{0:t})$. In the original FastSLAM algorithm, the map is feature-based and each feature is tracked using a separate Kalman filter, i.e., each particle contains a separate mean and covariance matrix for each feature in the map (making the map distribution $p(Y | x_{0:t}, z_{0:t})$ a density). The fact that each feature can be tracked separately means that the number of parameters only grows linearly with the number of features. As a result, the algorithm scales very well to large environments. Note that directly estimating the joint distribution $p(x_{0:t}, Y | z_{0:t}, u_{1:t})$ using a particle filter would be impossible because the number of particles required for a robust estimate usually grows exponentially with the number of dimensions and the map is extremely high-dimensional (the number of features/cells can easily be in the thousands or millions).

Even though FastSLAM estimates the distribution over the entire path, the actual algorithm works recursively and, in each time step, only the previous pose x_{t-1} is considered. Thus, if one is not interested in the entire path, one can simply drop previous poses and force particles not to grow over time. However, from a theoretical standpoint, the algorithm nonetheless estimates the entire path, which can also cause problems. In particular when the robot drives a large loop, the algorithm cannot effectively use measurements to correct for localization errors and has to rely almost entirely on the control information in order to update the pose (i.e., the pose error does accumulate over time). Thus, the algorithm can only successfully “close” the loop if at least one particle reflects the true path traveled during the loop, which becomes increasingly less likely with a growing loop size. This is one of the reasons why an extended version of FastSLAM was proposed in [Montemerlo et al., 2003] where not only the controls but also the measurements are used to predict subsequent poses. This extension is not further considered in this chapter, although it is a problem worth studying in the future.

In [Hähnel et al., 2003] and [Eliazar and Parr, 2003], the FastSLAM algorithm was applied to an occupancy grid map representation. This adaptation is rather straightforward because it simply means that each particle contains an occupancy grid map opposed to a set of Kalman filter estimates.

4.3. Evidential FastSLAM

In this section, the FastSLAM algorithm is generalized in the belief function framework. This means, instead of the joint probability distribution defined by Eq. (4.7), the joint belief function $m_{\Theta_{X_{0:t}} \times \Theta_Y^M}[z_{0:t}, u_{1:t}]$ is considered where $\Theta_{X_{0:t}}$ denotes the space of all paths up to time t . In order to make computing this belief function feasible, one assumption is made: the marginal distribution over the path is Bayesian.²

$$m_{\Theta_{X_{0:t}} \times \Theta_Y^M}^{\downarrow \Theta_{X_{0:t}}}[z_{0:t}, u_{1:t}](x_{0:t}) = p(x_{0:t}|z_{0:t}, u_{1:t}), \quad \forall x_{0:t} \in \Theta_{X_{0:t}} \quad (4.9)$$

There are three reasons for this assumption:

- The motion of a robot can usually be accurately described by a probabilistic model (usually by an additive Gaussian noise model, see Sect. 4.4.1).
- Continuous state variables are generally problematic in the belief function framework (see Sect. 3.7).
- Without the assumption, computing the joint belief function would be practically infeasible because it would not be possible to exploit conditional independence of grid cells given the path. (Each path hypothesis would actually correspond to a set of paths in this case.)

Using the assumption, the joint mass function $m_{\Theta_{X_{0:t}} \times \Theta_Y^M}[z_{0:t}, u_{1:t}]$ over path and map can be factorized into a probabilistic localization problem and a conditional evidential mapping problem.

$$m[z_{0:t}, u_{1:t}](x_{0:t}, Y) = p(x_{0:t}|z_{0:t}, u_{1:t}) m[x_{0:t}, z_{0:t}](Y) \quad (4.10)$$

This factorization follows from a generalized version of the product rule for probabilities and resembles the factorization underlying the classical FastSLAM algorithm. In Sect. A.2, a proof is given for this generalized product rule. Like in Eq. (4.7), the controls can be omitted for the map belief because they are conditionally independent given the path. The (approximate) conditional independence of grid cells also holds because they are being conditioned on the entire path $x_{0:t}$.

²Technically, the joint distribution $m_{\Theta_{X_{0:t}} \times \Theta_Y^M}[z_{0:t}, u_{1:t}]$ is a belief *density* function because the path is real-valued. However, this fact can be ignored here because the path is modeled probabilistically.

Like in the original FastSLAM algorithm, the joint distribution is approximated using a Rao-Blackwellized particle filter where each particle represents an entire path and a corresponding map belief function, which is updated analytically. The localization problem is in fact only partially probabilistic because only the path prior and posterior are assumed to be Bayesian. The next two sections describe localization and mapping in detail and, in Sect. 4.3.3, the resulting evidential FastSLAM algorithm is presented.

4.3.1. Localization

The localization of the robot turns out to be quite similar to classical Markov localization [Thrun et al., 2001] because the path is modeled probabilistically. The path probability distribution $p(x_{0:t}|z_{0:t}, u_{1:t})$ is computed recursively over time by performing a prediction step to model state changes and by performing a correction step to incorporate new measurements. This results in equations similar to the ones underlying Bayesian filtering shown in Sect. 3.3.3.

Prediction Step

In the prediction step, the path distribution at time t is computed from the path distribution at time $t - 1$ using the robot's motion model. This motion model describes how the robot's pose changes based on control u_t (an example of a typical 2D motion model is presented in Sect. 4.4.1). Because the underlying process is Markovian, only the previous pose has to be considered for the motion model:

$$p(x_t|x_{t-1}, u_t). \quad (4.11)$$

The prior $p(x_{0:t-1}|z_{0:t-1}, u_{1:t-1})$ can simply be updated by applying the product rule and by exploiting the fact that the motion model only depends on the previous pose x_{t-1} and the latest control u_t . The resulting distribution $p(x_{0:t}|z_{0:t-1}, u_{1:t})$, which includes the latest control u_t but not the latest measurement z_t , forms the proposal distribution.

$$p(x_{0:t}|z_{0:t-1}, u_{1:t}) = p(x_t|x_{t-1}, u_t) p(x_{0:t-1}|z_{0:t-1}, u_{1:t-1}) \quad (4.12)$$

Note that the initial prior $p(x_0)$ simply assigns all mass to a single point because, without a map, it does not make sense to express pose uncertainty.

Correction Step

Incorporating a new measurement z_t probabilistically would consist of an application of the classical Bayesian theorem where the proposal distribution $p(x_{0:t}|z_{0:t-1}, u_{1:t})$ is multiplied with the likelihood of the new measurement in order to obtain the posterior. However, the constraint defined by Eq. (4.9) only requires that the path distributions at time $t - 1$ and t are Bayesian. It does

not specify the way in which the prior is updated, i.e., it does not require an application of the classical Bayesian theorem. As a result, one is free to assume an arbitrary belief function for the measurement likelihood and it suffices to assume that the initial prior $p(x_0)$ and the motion model $p(x_t|x_{t-1}, u_t)$ are Bayesian in order to satisfy Eq. (4.9) (see also Sect. 3.3.3). In this case, the correction step consists of an application of the generalized Bayesian theorem with a Bayesian prior and posterior.

$$p(x_{0:t}|z_{0:t}, u_{1:t}) \propto pl[x_{0:t}, z_{0:t-1}](z_t) p(x_{0:t}|z_{0:t-1}, u_{1:t}) \quad (4.13)$$

This equation can be derived as follows:

$$p(x_{0:t}|z_{0:t}, u_{1:t}) = (m_{\Theta_{x_{0:t}}}[z_{0:t}, u_{1:t}] \oplus p(\cdot|z_{0:t-1}, u_{1:t}))(x_{0:t}) \quad (\text{Eq. (4.9)}) \quad (4.14)$$

$$\propto pl[z_{0:t}, u_{1:t}](x_{0:t}) p(x_{0:t}|z_{0:t-1}, u_{1:t}) \quad (\text{Proof A.1}) \quad (4.15)$$

$$\propto pl[x_{0:t}, z_{0:t-1}, u_{1:t}](z_t) p(x_{0:t}|z_{0:t-1}, u_{1:t}) \quad (\text{Eq. (2.72)}) \quad (4.16)$$

$$\propto pl[x_{0:t}, z_{0:t-1}](z_t) p(x_{0:t}|z_{0:t-1}, u_{1:t}) \quad (\text{Cond. Ind.}) \quad (4.17)$$

Eq. (4.9) requires that the prior and the posterior are Bayesian (the fact that they are densities is ignored here). When applying the generalized Bayesian theorem, the prior is fused using Dempster's rule of combination (see Eq. (2.75)), resulting in Eq. (4.14). Note that the underlying evidence is distinct because $m_{\Theta_{x_{0:t}}}[z_{0:t}, u_{1:t}]$ is computed using the generalized Bayesian theorem which only utilizes the likelihood of z_t . Because the prior is Bayesian, the combination is also Bayesian (the proof is very simple and can be found in [Shafer, 1976, chapter 3]). In Sect. A.1, it is proved that the combination of two mass functions where one of the mass functions is Bayesian can be expressed as a product of a probability and a plausibility (Eq. (4.15)). Here, the generalized Bayesian theorem defined by Eq. (2.72) is applied, resulting in the plausibility $pl[x_{0:t}, z_{0:t-1}, u_{1:t}](z_t)$ in Eq. (4.16). Finally, conditional independence of measurement z_t with respect to the controls given the path is exploited in Eq. (4.17). Note that z_t is *not* conditionally independent of the remaining measurements because of the unknown map (this can also be seen in Fig. 4.1 where measurements do not only depend on the poses but also on the map).

The important part in Eq. (4.13) is $pl[x_{0:t}, z_{0:t-1}](z_t)$ which states how likely measurement z_t is given all previous measurements and poses. At first sight, this distribution looks very difficult to compute because of the dependence on past measurements and poses and it appears to be unsuited for recursive updating. However, in FastSLAM, each particle also contains a distribution describing the map, which is why the likelihood can be conditioned on the map using the f -total law. In this case, the current measurement z_t indeed becomes conditionally independent of past measurements and poses.

$$pl[x_{0:t}, z_{0:t-1}](z_t) \stackrel{(2.63)}{=} \sum_{Y \subseteq \Theta_Y^M} pl[x_t, Y](z_t) m[x_{0:t-1}, z_{0:t-1}](Y) \quad (4.18)$$

Here, the distribution $pl[x_t, Y](z_t)$ represents the forward sensor model while $m[x_{0:t-1}, z_{0:t-1}](Y)$ represents the map belief at time $t - 1$. The sum over Y may seem intractable because of the iteration over the power set of the space of all possible maps Θ_Y^M (note that $|\mathcal{P}(\Theta_Y^M)| = 2^{2^M}$ where usually $M > 10,000$). Fortunately, the independence assumption between grid cells reduces the complexity to $O(M)$ and, in Sect. 4.4.2, an efficient implementation is presented.

4.3.2. Grid Mapping

Like the path distribution, the map distribution $m_{\Theta_Y^M}[x_{0:t}, z_{0:t}]$ can be updated recursively over time because each new measurement z_t is conditionally independent of all past measurements and past poses given the current pose x_t and the map. Therefore, the prior belief $m_{\Theta_Y^M}[x_{0:t-1}, z_{0:t-1}]$ can be combined with the measurement-induced belief $m_{\Theta_Y^M}[x_t, z_t]$ using an appropriate combination rule $\otimes \in \{\odot, \oplus, \dots\}$ (different combination rules for mapping are compared in Sect. 4.5).

$$m_{\Theta_Y^M}[x_{0:t}, z_{0:t}] = m_{\Theta_Y^M}[x_{0:t-1}, z_{0:t-1}] \otimes m_{\Theta_Y^M}[x_t, z_t] \quad (4.19)$$

The mass function $m_{\Theta_Y^M}[x_t, z_t]$ represents the inverse sensor model (see Sect. 4.4.3) which provides a map distribution derived from a single measurement.

As argued above, grid cells can be considered as approximately independent of each other if conditioned on the entire path like in Eq. (4.10). As a result, the joint belief distribution $m_{\Theta_Y^M}[x_{0:t}, z_{0:t}]$ over all grid cells can be expressed as M marginal mass functions that are combined using Dempster's rule. Equivalently, the joint belief distribution can be factorized and expressed as a product (similar to the probabilistic version shown in Eq. 4.8).

$$m_{\Theta_Y^M}[x_{0:t}, z_{0:t}] = \bigoplus_{i=1}^M m_{\Theta_{Y;i}}[x_{0:t}, z_{0:t}] \quad (4.20)$$

$$m[x_{0:t}, z_{0:t}](Y_{1:M}) = \prod_{i=1}^M m[x_{0:t}, z_{0:t}](Y_i) \quad (4.21)$$

The independence assumption implies that each mass function $m[x_{0:t}, z_{0:t}](Y_i)$ has only up to four focal sets (three, if the mass functions are normalized). As a consequence, only $3M$ parameters have to be estimated during mapping opposed to $2^{2^M} - 1$ parameters. Note that for mapping, mass functions are generally assumed to be unnormalized. The reason for this assumption is that unnormalized mass functions contain an additional parameter for each grid cell (the mass assigned to \emptyset) which provides additional information about the amount of conflict between measurements both over time and across different sensors.³ In contrast, the mass assigned to Θ_Y represents a lack of evidence or ignorance regarding the

³This can also be interpreted as a form of open-world assumption [Smets, 1988] because the model allows for the possibility that a cell is neither strictly occupied nor strictly empty.

cell's state. Thus, compared to classical grid mapping, the evidential mapping approach provides two additional parameters for representing uncertainty about each cell.

Because the grid cell decomposition defined by Eq. (4.20) also applies to the inverse sensor model, the map update defined by Eq. (4.19) can be expressed in terms of the recursive update of a single cell.

$$m_{\Theta_Y}[x_{0:t}, z_{0:t}] = m_{\Theta_Y}[x_{0:t-1}, z_{0:t-1}] \circledast m_{\Theta_Y}[x_t, z_t] \quad (4.22)$$

Note that the cell prior m_{Θ_Y} at time $t = 0$ is assumed to be vacuous for all cells (unless there already exists a partial map or some other prior knowledge).

$$m(Y_i = \Theta_Y) = 1, \quad 1 \leq i \leq M \quad (4.23)$$

This is an advantage compared to probabilistic grid mapping where one has to specify a Bayesian occupancy prior (usually $P(o) = 0.5$) even if nothing is known about the environment.

4.3.3. Algorithm

The evidential FastSLAM algorithm uses a particle filter to maintain a set of path hypotheses and corresponding map distributions. The main difference with respect to the original FastSLAM algorithm is that the forward and inverse sensor models are general belief functions and, as a result, each map distribution is also a belief function. The particle set \mathcal{X}_t at time t consists of K tuples $(x_{0:t}^{[k]}, m_{\Theta_Y^M}[x_{0:t}^{[k]}, z_{0:t}])$ with $1 \leq k \leq K$ where $x_{0:t}^{[k]}$ represents the k -th path hypothesis and $m_{\Theta_Y^M}[x_{0:t}^{[k]}, z_{0:t}]$ represents the corresponding map belief. Like the particle filter algorithm presented in Sect. 3.4, the evidential FastSLAM algorithm is based on importance sampling and updates the particle set recursively over time by performing a prediction step, an importance weighting step, and a subsequent resampling step. In addition, a map update is performed for each measurement using an appropriate combination rule.

The resulting algorithm is shown in Fig. 4.2. In the following, the four steps performed by the algorithm are described in more detail:

1. Prediction step (line 3): Sample a new pose $x_t^{[k]}$ from the motion model $p(x_t|x_{t-1}^{[k]}, u_t)$ in order to update the robot's current state and incorporate control u_t .
2. Importance weighting (line 4): Compute weight $w_t^{[k]} = pl[x_{0:t}^{[k]}, z_{0:t-1}](z_t)$ for each particle using the forward sensor model $pl[x_t, Y](z_t)$ and the *current* map belief $m_{\Theta_Y^M}[x_{0:t-1}^{[k]}, z_{0:t-1}]$, i.e., the one without the latest measurement z_t incorporated, see Eq. (4.18). This process is described in Sect. 4.4.2. The resulting importance weights are added to the weighted particle set $\tilde{\mathcal{X}}_t$ in line 6 along with the predicted path $x_{0:t}^{[k]}$ and the updated map (next step).

Algorithm: Evidential FastSLAM	
Input: $\mathcal{X}_{t-1}, z_t, u_t$ // prior sample set, measurement, control	
1 $\tilde{\mathcal{X}}_t \leftarrow \emptyset$	// weighted sample set
2 for $k \leftarrow 1$ to K do	
3 sample $x_t^{[k]} \sim p(x_t x_{t-1}^{[k]}, u_t)$	// sample pose
4 $w_t^{[k]} \leftarrow pl[x_{0:t}^{[k]}, z_{0:t-1}](z_t)$	// compute importance weight
5 $m_{\Theta_Y^M}[x_{0:t}^{[k]}, z_{0:t}] \leftarrow m_{\Theta_Y^M}[x_{0:t-1}^{[k]}, z_{0:t-1}] \circledast m_{\Theta_Y^M}[x_t^{[k]}, z_t]$	// update map
6 add $(x_{0:t}^{[k]}, m_{\Theta_Y^M}[x_{0:t}^{[k]}, z_{0:t}], w_t^{[k]})$ to $\tilde{\mathcal{X}}_t$	
7 end	
8 $\mathcal{X}_t \leftarrow \emptyset$	// updated sample set
9 for $k \leftarrow 1$ to K do	// importance resampling
10 draw $(x_{0:t}^{[k]}, m_{\Theta_Y^M}[x_{0:t}^{[k]}, z_{0:t}])$ from $\tilde{\mathcal{X}}_t$ with probability $\propto w_t^{[k]}$	
11 add $(x_{0:t}^{[k]}, m_{\Theta_Y^M}[x_{0:t}^{[k]}, z_{0:t}])$ to \mathcal{X}_t	
12 end	
13 return \mathcal{X}_t	

Figure 4.2.: Evidential FastSLAM algorithm. The updated particle set \mathcal{X}_t is constructed from the previous particle set \mathcal{X}_{t-1} , the latest measurement z_t , and the latest control u_t .

3. Map update (line 5): Update the current map belief $m_{\Theta_Y^M}[x_{0:t-1}^{[k]}, z_{0:t-1}]$ using the inverse sensor model $m_{\Theta_Y^M}[x_t^{[k]}, z_t]$ (described in Sect. 4.4.3) and an appropriate combination rule.
4. Resampling (lines 9–12): Resample particles from the weighted set $\tilde{\mathcal{X}}_t$ with probability proportional to the importance weights. This results in the final (unweighted) set \mathcal{X}_t representing the joint path/map belief $m_{\Theta_{x_{0:t}} \times \Theta_Y^M}[z_{0:t}, u_{1:t}]$ reflecting all measurements and controls up to time t .

The time complexity of the evidential FastSLAM algorithm is $O(KM)$, assuming that the complexity of evaluating the sensor models is $O(M)$ (the models presented in the next sections indeed have this property). This is because the algorithm has to iterate over all particles as well as over all cells. The complexity therefore does not differ from probabilistic FastSLAM for grid mapping (there is a constant overhead caused by the fact that each cell is represented by three parameters instead of one).

4.4. Models

This section presents implementations of the different models required by the evidential FastSLAM algorithm. There are three models that need to be provided

to the algorithm:

- A motion model $p(x_t|x_{t-1}, u_t)$ predicting the next pose based on the current pose and control u_t .
- A forward sensor model $pl[x_t, Y](z_t)$ specifying the plausibility of a measurement given the current pose x_t and the map Y . This model is used to compute the plausibility $pl[x_{0:t}, z_{0:t-1}](z_t)$ of the current measurement z_t given all previous poses and measurements.
- An inverse sensor model $m[x_t, z_t](Y)$ providing a local map belief derived from a single measurement in order to update the global map belief $m[x_{0:t-1}, z_{0:t-1}](Y)$.

The forward and inverse sensor models directly depend on the type of sensor (e.g., camera, laser scanner, or sonar). Here, the focus is on sonar which provides range measurements. The advantage of sonar is that it is a widely-available low-cost sensor and that range measurements can be easily incorporated into an occupancy grid map. Furthermore, sonar sensors exhibit significant errors, which makes them interesting from a modeling standpoint, both regarding the forward model and the inverse model (although existing evidential approaches for sonar usually focus exclusively on the inverse model, e.g., [Yang and Aitken, 2006, Gambino et al., 1997]).

The remainder of this section introduces a 2D motion model, a sonar forward model, and a sonar inverse model. The forward model is first defined for the case of a known map and then for the case where the map is described by a belief function. The inverse sensor model is directly derived from the forward model.

4.4.1. Motion Model

The motion model describes how the robot's pose changes when executing a control. For the 2D case, Eq. (4.2) defines a pose x_t as a pair of Cartesian coordinates $x_{t;x}, x_{t;y}$ and an angle $x_{t;\phi}$. A control u_t consists of a velocity $u_{t;v}$ in the direction of the current bearing and an angular velocity $u_{t;w}$ as defined by Eq. (4.6). By means of simple geometry, a function g can be defined which, ignoring noise, computes pose x_t from the previous pose x_{t-1} and control u_t where Δt denotes the amount of time between two consecutive time steps.

$$g(x_{t-1}, u_t) = x_{t-1} + \begin{pmatrix} \cos(x_{t-1;\phi} + u_{t;w} \Delta t) u_{t;v} \Delta t \\ \sin(x_{t-1;\phi} + u_{t;w} \Delta t) u_{t;v} \Delta t \\ u_{t;w} \Delta t \end{pmatrix} \quad (4.24)$$

Note that this model assumes that rotation and translation do not occur simultaneously and that rotation occurs *before* translation. Even if translation and rotation do occur simultaneously, the model still works as a good approximation

if Δt is sufficiently small with respect to the velocities (in Sect. 4.5, Δt is very small with $\Delta t \approx 15ms$).

The actual pose x_t is assumed to be the result of the geometric function g and an additive noise term ϵ_u . The noise term is assumed to be normally distributed with 0 mean and a diagonal covariance matrix Σ_u (the amount of noise is assumed to independent of the velocity).

$$x_t = g(x_{t-1}, u_t) + \epsilon_u \quad (4.25)$$

$$\epsilon_u \sim \mathcal{N}(0, \Sigma_u) \quad (4.26)$$

$$\Sigma_u = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\phi^2 \end{pmatrix} \Delta t \quad (4.27)$$

The motion model $p(x_t|x_{t-1}, u_t)$ is thus defined as the following Gaussian:

$$p(x_t|x_{t-1}, u_t) = \mathcal{N}(x_t; g(x_{t-1}, u_t), \Sigma_u). \quad (4.28)$$

For the experiments in Sect. 4.5, the values $\sigma_x = \sigma_y \approx 1.6 \cdot 10^{-2}m$ and $\sigma_\phi \approx 0.48^\circ$ are used. Note that this is a very simple motion model and a more elaborate model could reduce estimation errors in practice.

4.4.2. Forward Sensor Model

There are two cases in which the plausibility of the current measurement z_t is computed: the forward sensor model $pl[x_t, Y](z_t)$ conditioned on map Y and the importance weight plausibility $pl[x_{0:t}, z_{0:t-1}](z_t)$ conditioned on all previous poses and measurements. These two cases are analyzed in the following.

Conditioned on Map

The forward sensor model $pl[x_t, Y](z_t)$ specifies how plausible a range measurement z_t is given the current pose x_t and the map $Y \subseteq \Theta_Y^M$ (which is actually a set of maps because Y is an evidential variable).

The uncertainty of the measurement process is modeled as the result of two effects:

- Noise: sonar measurements are generally noisy which is modeled using an additive Gaussian error term (just like in the motion model).
- Erraticness/randomness: sonar measurements are sometimes entirely random. While this is in part an inherent property of sonar, it also captures unpredictable events caused by model limitations (e.g., a person walking in front of the sensor). Here, such measurements are referred to as *random*.

First, random measurements are modeled. Let $R \subseteq \Theta_R = \{r, \neg r\}$ be a binary evidential variable representing whether a measurement is random where r represents *random*. Let ϵ_r with $0 \leq \epsilon_r < 1$ denote the maximum chance of receiving such a random measurement. This corresponds to a simple belief function over Θ_R with

$$m(\neg r) = 1 - \epsilon_r, \quad (4.29)$$

$$m(\Theta_R) = \epsilon_r. \quad (4.30)$$

This means that $1 - \epsilon_r$ is the minimum chance of receiving a non-random measurement and ϵ_r represents ignorance regarding the possibility of a random measurement.

If a measurement is entirely random, any value in Ω_t is completely plausible.

$$pl[x_t, Y, r](z_t) = 1, \quad \forall z_t \in \Omega_t \quad (4.31)$$

As a result, any measurement is also completely plausible if it is unknown whether the measurement is random (i.e., when conditioning on the disjunction Θ_R). This follows directly from an application of the disjunctive rule of combination.

$$pl[x_t, Y, \Theta_R](z_t) \stackrel{(2.62)}{=} 1 - \prod_{R \in \Theta_R} (1 - pl[x_t, Y, R](z_t)) \stackrel{(4.31)}{=} 1 \quad (4.32)$$

Intuitively, this makes sense because Θ_R includes the random case, which means that it is entirely plausible for the measurement to take on any value $z_t \in \Omega_t$.

Finally, by conditioning the forward model on R using the f -total law, one obtains a model for random measurements which simply states that any measurement has a plausibility of at least ϵ_r .

$$pl[x_t, Y](z_t) \stackrel{(2.63)}{=} \sum_{R \subseteq \Theta_R} pl[x_t, Y, R](z_t) m(R) \quad (4.33)$$

$$= (1 - \epsilon_r) pl[x_t, Y, \neg r](z_t) + \epsilon_r \quad (4.34)$$

Note that there is no normalization for the conditioning (see Eq. (2.64)) because R is defined over a separate frame of discernment. Also note that conditioning on R is equivalent to discounting with factor $1 - \epsilon_r$ [Shafer, 1976, chapter 11].

Aside from complete randomness, additive noise is the other major source of uncertainty for sonar measurements. Before proceeding with modeling this noise, some additional notation needs to be introduced. Let M' denote the number of grid cells located inside the measurement cone ($M' \leq M$). Furthermore, let $Y'_{1:M'}$ with $Y'_i \subseteq \Theta_Y$ denote the sequence of grid cells located inside the measurement cone sorted in ascending order by their Euclidean distance from the robot's current position (i.e., $Y'_{M'}$ is the most distant cell inside the measurement cone). By definition, cells outside of the measurement cone do not influence the measurement and can therefore be ignored.

Let C be an evidential variable representing the cause of a measurement with frame of discernment $\Theta_C = \{1, \dots, M', M' + 1\}$. The value $M' + 1$ represents the case where all cells are empty and the sensor returns the maximum range z_{\max} . All other values $c \in \Theta_C$ represent the case where cell Y'_c is the cause of the measurement. It is assumed that there always is exactly one cause. Like for randomness, the forward model $pl[x_t, Y, \neg r](z_t)$ can be conditioned on C .

$$pl[x_t, Y, \neg r](z_t) \stackrel{(2.63)}{=} \sum_{C \subseteq \Theta_C} pl[x_t, \neg r, C](z_t) m[Y'_{1:M'}](C) \quad (4.35)$$

Here, the map can be omitted for $pl[x_t, \neg r, C](z_t)$ because the map is conditionally independent of the measurement if the cause C is given. For the distribution of C , only the sequence of cells inside the measurement cone is relevant.

If C is a singleton, the measurement plausibility can be directly specified. For $c \leq M'$, the c -th cell contains the measured obstacle and the measurement z_t should be the distance to that cell with added Gaussian noise. For $c = M' + 1$, the plausibility is 1 at z_{\max} and 0 everywhere else.

$$pl[x_t, c, \neg r](z_t) = \begin{cases} \alpha \mathcal{N}(z_t; \mu_c, \sigma_z^2) & \text{if } c \leq M', \\ 1 & \text{if } c = M' + 1 \wedge z_t = z_{\max}, \\ 0 & \text{else} \end{cases} \quad (4.36)$$

The mean μ_c is the distance to the c -th cell. For the noise parameter, $\sigma_z = 0.125m$ is assumed. The normalization constant α asserts that the Gaussian is bounded by 1 because it is treated as a plausibility.

If C is not a singleton, the disjunctive rule of combination can be applied to construct the plausibility for arbitrary subsets $C \subseteq \Theta_C$ using the singleton plausibility defined by Eq. (4.36).

$$pl[x_t, C, \neg r](z_t) \stackrel{(2.62)}{=} 1 - \prod_{c \in C} (1 - pl[x_t, c, \neg r](z_t)) \quad (4.37)$$

Next, the cause distribution $m[Y'_{1:M'}](C)$ needs to be specified. An ideal range sensor would always measure the distance to the closest obstacle located inside the measurement cone, i.e., the distance to the closest occupied cell. While, in reality, this assumption is not generally true, it is made here nonetheless in order to make computing the forward model feasible. Using this assumption, the distribution of C is greatly simplified because, in this case, it is a categorical belief function.

$$m[Y'_{1:M'}](C_{Y'}^*) = 1 \quad (4.38)$$

$$C_{Y'}^* = \{c | c \in \Theta_C, o \in Y'_c, (\neg \exists i : i < c \wedge Y'_i = \{o\})\} \quad (4.39)$$

The set $C_{Y'}^*$ simply contains all potentially occupied cells ($o \in Y'_c$). Furthermore, if the c -th cell is strictly occupied ($Y'_c = \{o\}$), all cells that are farther away can

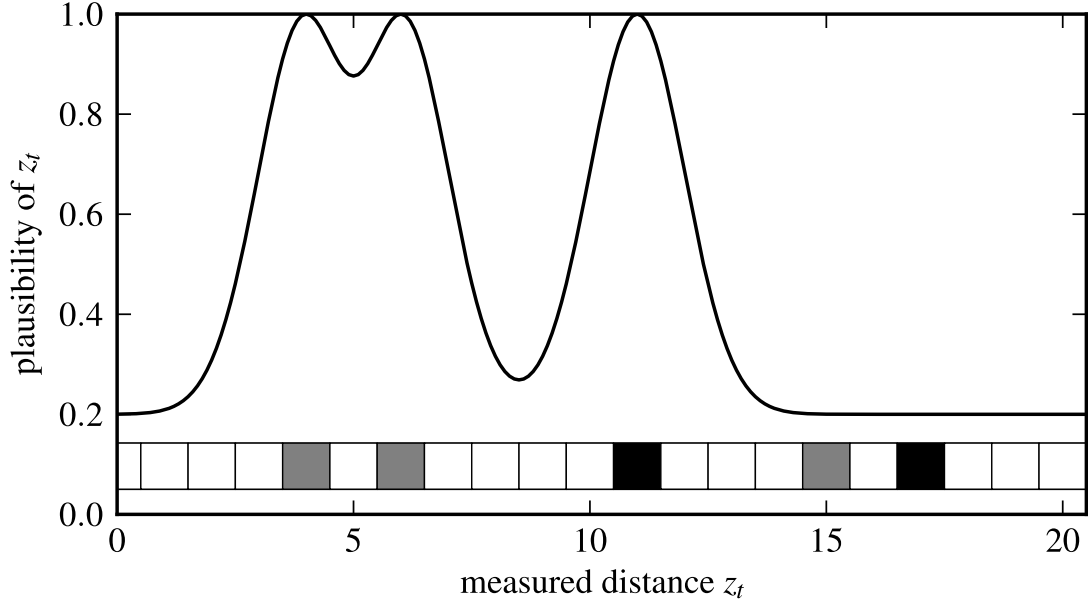


Figure 4.3.: Forward sensor model $pl[x_t, Y](z_t)$ as a function of measurement z_t for a 1D map. The squares at the bottom represent the grid cells: white means $\{-o\}$, black means $\{o\}$, and gray means Θ_Y . (Figure adopted from [Reineking and Clemens, 2013].)

be ignored as potential causes because of the simplifying assumption of always measuring the closest occupied cell. Without this assumption, e.g., by allowing the measurement beam to miss an occupied cell with a certain probability, the distribution over Θ_C could become arbitrarily complex with up to $2^{M'} - 1$ focal sets where M' can easily be in the thousands. Note that $Y'_{M'+1}$ represents a “virtual cell” at the end of the measurement cone that is always strictly occupied and, as a result, is always hit if all other cells are empty.

Finally, by combining Eq. (4.34) with (4.38), the complete forward sensor model is obtained.

$$pl[x_t, Y](z_t) = (1 - \epsilon_r) \left(1 - \prod_{c \in C_{Y'}^*} (1 - pl[x_t, c, \neg r](z_t)) \right) + \epsilon_r \quad (4.40)$$

Fig. 4.3 shows an example of the forward model for an artificial 1D map. For simplicity, cell indices are assumed to coincide with cell distances (i.e., $\mu_i = i$). The set $C_{Y'}^*$ contains three indices with $C_{Y'}^* = \{4, 6, 11\}$ in this example (the first two gray cells Y'_4 and Y'_6 representing Θ_Y and the first strictly occupied cell Y'_{11}). For all three cells, the model exhibits Gaussian peaks at the corresponding distances according to Eq. (4.36). The (potentially) non-empty cells Y'_{15} and Y'_{17} do not cause the likelihood to peak because they are not included in $C_{Y'}^*$, due to the strictly occupied cell Y'_{11} . The plausibility function is bounded below because of $\epsilon_r = 0.2$ which means that any z_t has at least a certain amount of plausibility.

Note that in case measurement z_t is a vector $z_t = (z_{t;1}, \dots, z_{t;N})^T$ composed of conditionally independent range measurements, the forward sensor model can be expressed as a product of the plausibilities corresponding to each component $z_{t;l}$ because of conditional cognitive independence.

$$pl[x_t, Y](z_t) \stackrel{(2.69)}{=} \prod_{l=1}^N pl[x_t, Y](z_{t;l}) \quad (4.41)$$

Here, each plausibility $pl[x_t, Y](z_{t;l})$ is computed according to Eq. (4.40).

Conditioned on Previous Poses and Measurements

For computing the importance weights, the map is not given and the expression $pl[x_{0:t}, z_{0:t-1}](z_t)$ needs to be computed by conditioning on the map as shown in Eq. (4.18), which is restated here for convenience:

$$pl[x_{0:t}, z_{0:t-1}](z_t) = \sum_{Y \subseteq \Theta_Y^M} pl[x_t, Y](z_t) m[x_{0:t-1}, z_{0:t-1}](Y).$$

The map belief $m[x_{0:t-1}, z_{0:t-1}](Y)$ is known because each particle contains its own mass function for the map. Though the map belief can be unnormalized depending on the combination rule used in Eq. (4.19), for localization, only the normalized map belief is considered. Without normalization, maps with lower conflict would generally be favored during importance sampling. However, in practice, this is not always desirable because maps with high conflict sometimes more accurately represent the true map.

Modeling entirely random measurements is done in the same way like in the known-map case by conditioning on R as shown in Eq. (4.34).

$$pl[x_{0:t}, z_{0:t-1}](z_t) = (1 - \epsilon_r) pl[x_{0:t}, z_{0:t-1}, \neg r](z_t) + \epsilon_r \quad (4.42)$$

To simplify notation, define $m(Y') := m[x_{0:t-1}, z_{0:t-1}](Y')$ where $Y' = Y'_{1:M'+1}$ (only the part of the map inside the measurement cone needs to be considered). Here, $Y'_{M'+1}$ represents the “virtual cell” at the end of the measurement cone with $m(Y'_{M'+1} = \{o\}) = 1$. The plausibility of z_t is obtained by conditioning on the partial map Y' as defined by Eq. (4.18) and on the cause variable C as shown in Eq. (4.35). Furthermore, by exploiting the fact that cells are assumed to be independent, the joint cell belief $m(Y')$ can be factorized as shown in Eq. (4.21) and the sum over subsets of the joint space can be expressed in terms of a separate sum for each cell.

$$pl[x_{0:t}, z_{0:t-1}, \neg r](z_t) \quad (4.43)$$

$$= \sum_{Y' \subseteq \Theta_{Y'}^{M'}} \sum_{C \subseteq \Theta_C} pl[x_t, \neg r, C](z_t) m[Y'](C) m(Y') \quad (4.44)$$

$$= \sum_{Y'_1 \subseteq \Theta_{Y'_1}} m(Y'_1) \cdots \sum_{Y'_{M'+1} \subseteq \Theta_{Y'_{M'+1}}} m(Y'_{M'+1}) pl[x_t, \neg r, C_{Y'}^*](z_t) \quad (4.45)$$

Because $m[Y'](C)$ is categorical as defined by Eq. (4.38), the sum over C can be omitted and only the focal set $C_{Y'}^*$ needs to be considered.

If the first $i - 1$ cells are known to be empty (denoted by $Y'_{1:i-1} = \{\neg o\}^{i-1}$), the first $i - 1$ sums disappear in Eq. (4.45) because the corresponding cells are by definition not part of $C_{Y'}^*$.

$$\begin{aligned} & pl[x_{0:t}, z_{0:t-1}, \neg r, Y'_{1:i-1} = \{\neg o\}^{i-1}](z_t) \\ &= \sum_{Y'_i \subseteq \Theta_Y} m(Y'_i) \cdots \sum_{Y'_{M'+1} \subseteq \Theta_Y} m(Y'_{M'+1}) pl[x_t, C_{Y'}^*](z_t) \end{aligned} \quad (4.46)$$

This equation can be computed recursively over the sequence of cells inside the measurement cone. This becomes apparent if the possible assignments to cell Y'_i are considered.

$$\begin{aligned} & pl[x_{0:t}, z_{0:t-1}, \neg r, Y'_{1:i-1} = \{\neg o\}](z_t) \\ &= \sum_{Y'_i \subseteq \Theta_Y} m(Y'_i) \cdot \begin{cases} pl[x_t, \neg r, C_{Y'}^* = \{i\}](z_t) & \text{if } Y'_i = \{o\} \\ pl[x_{0:t}, z_{0:t-1}, \neg r, Y'_{1:i} = \{\neg o\}^i](z_t) & \text{if } Y'_i = \{\neg o\} \\ pl[x_{0:t}, z_{0:t-1}, \neg r, Y'_{1:i-1} = \{\neg o\}^{i-1}, \\ \quad Y'_i = \Theta_Y](z_t) & \text{if } Y'_i = \Theta_Y \end{cases} \end{aligned} \quad (4.47)$$

The case $Y'_i = \emptyset$ can be ignored because $m(Y'_i)$ is assumed to be normalized. The occupied case $Y'_i = \{o\}$ implies that $C_{Y'}^* = \{i\}$ which means that the plausibility is directly given by Eq. (4.36). For the empty case $Y'_i = \{\neg o\}$, the i -th cell can be ignored because it is not part of $C_{Y'}^*$, and the plausibility is obtained recursively with $Y'_{1:i} = \{\neg o\}^i$. The recursion is guaranteed to terminate because of the “virtual cell” $Y'_{M'+1}$ corresponding to a maximum range measurement. The expression for the case $Y'_i = \Theta_Y$ is a combination of the other two cases and it can be computed using the disjunctive rule of combination.

$$\begin{aligned} & pl[x_{0:t}, z_{0:t-1}, \neg r, Y'_{1:i-1} = \{\neg o\}^{i-1}, Y'_i = \Theta_Y](z_t) \\ & \stackrel{(2.62)}{=} 1 - \prod_{y'_i \in \Theta_Y} (1 - pl[x_{0:t}, z_{0:t-1}, \neg r, Y'_{1:i-1} = \{\neg o\}^{i-1}, y'_i](z_t)) \end{aligned} \quad (4.48)$$

Thus, the recursive algorithm shown in Fig. 4.4 can compute the likelihood $pl[x_{0:t}, z_{0:t-1}, \neg r](z_t)$ in $O(M')$ by going through all cells in order of their distance to the robot and computing the plausibilities for the cases $\{o\}$ and $\{\neg o\}$ where the first is directly given by Eq. (4.36) and the latter is obtained via recursion. The disjunctive case can simply be computed by storing the results of the previous two cases and by combining them according to Eq. (4.48).

Note that Eq. (4.47) can also be written without the sum and the case differentiation.

$$\begin{aligned} & pl[x_{0:t}, z_{0:t-1}, \neg r, Y'_{1:i-1} = \{\neg o\}^{i-1}](z_t) \\ &= pl[x_{0:t}, z_{0:t-1}, \neg r, Y'_{1:i} = \{\neg o\}^i](z_t) (pl(Y'_i = \{\neg o\}) - m(Y'_i = \Theta_Y)) \\ & \quad \cdot pl[x_t, \neg r, c = i](z_t) + pl(Y'_i = \{o\}) pl[x_t, \neg r, c = i](z_t) \end{aligned} \quad (4.49)$$

Function: importance_weight	
	Input: z_t, i // measurement, cell index
1	if $i = -1$ then
2	$pl \leftarrow \text{importance_weight}(z_t, 0)$
3	return $(1 - \epsilon_r) pl + \epsilon_r$ // random measurements
4	else if $i = M' + 1$ then
5	return $pl[x_t, c = i, \neg r](z_t)$ // virtual occupied cell
6	else
7	$pl_o \leftarrow pl[x_t, c = i, \neg r](z_t)$ // forward sensor model
8	$pl_{\neg o} \leftarrow \text{importance_weight}(z_t, i + 1)$ // recursion
9	$pl_{\Theta} \leftarrow pl_o + pl_{\neg o} - pl_o pl_{\neg o}$ // disjunctive rule
10	return $m(Y'_i = \{o\}) pl_o + m(Y'_i = \{\neg o\}) pl_{\neg o} + m(Y'_i = \Theta_Y) pl_{\Theta}$
11	end

Figure 4.4.: Recursive importance weight algorithm. The input of the function `importance_weight` is the measurement z_t and the cell index i . The function is initially called with $i = -1$ in order to handle the plausibility of random measurements separately. It then goes through all cells recursively and weighs the measurement plausibilities conditioned on the different subsets of Θ_Y with the corresponding cell beliefs $m(Y'_i)$.

Fig. 4.5 shows the plausibility $pl[x_{0:t}, z_{0:t-1}](z_t)$ for a 1D map where the map contains uncertainty (it is identical to the one shown in Fig. 4.3 except for the additional uncertainty). The effect of the uncertainty can be directly seen with the Gaussian peaks being only half as high because of the 0.5 mass on $\{\neg o\}$. For the same reason, the plausibility does not drop off to 0 behind cell Y'_{11} . Like in the forward model, the plausibility is bounded below due to ϵ_r . The small peak at the very right is caused by reaching z_{\max} (the plausibility does not go up to 1 here because of cells Y'_{11} and Y'_{17} each assigning mass 0.5 to $\{o\}$).

4.4.3. Inverse Sensor Model

The inverse sensor model $m[x_t, z_t](Y_i)$ introduced in Sect. 4.3.2 provides an occupancy belief distribution for each cell inside of the measurement cone (outside, the belief is vacuous). While several belief-based inverse models for sonar have been proposed in the literature [Yang and Aitken, 2006, Mullane et al., 2006, Gambino et al., 1997], the inverse model presented in this section is derived in a more principled manner from the forward model. This means that all the forward model parameters are reflected in the inverse model.

As in the case of the forward model, the cells $Y'_{1:M'}$ inside the measurement cone are assumed to be sorted by their distance with respect to the robot in ascending order. Unfortunately, it is not possible to obtain the inverse model

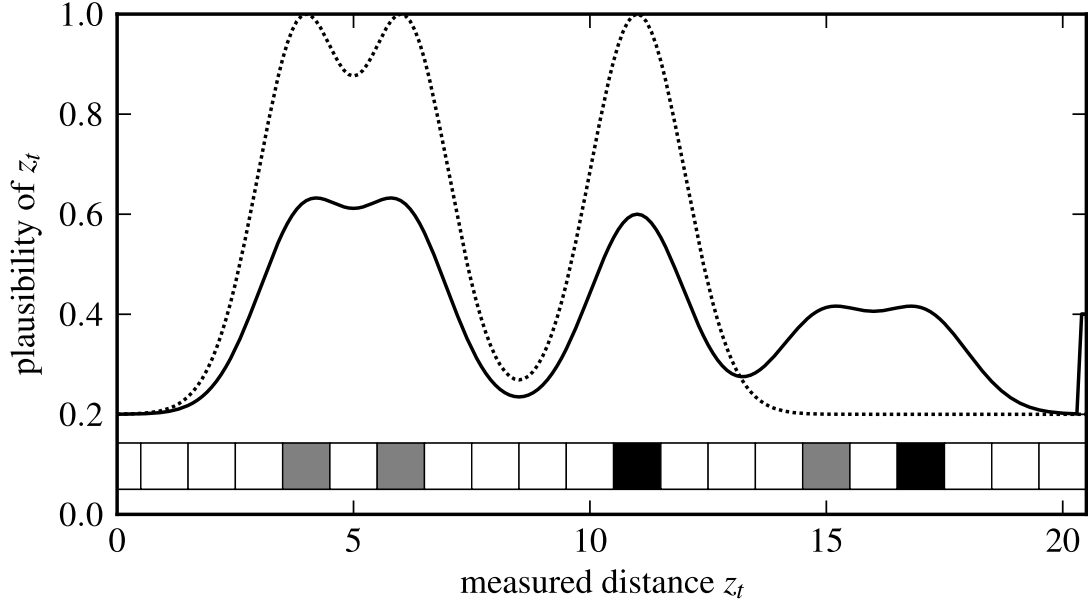


Figure 4.5.: Measurement plausibility $pl[x_{0:t}, z_{0:t-1}](z_t)$ (solid line) as a function of the measured distance z_t . The map indicated by the squares at the bottom is the same like in Fig. 4.3 and the dashed line shows the corresponding forward model plausibility. The difference with respect to Fig. 4.3 is that the map is described by a belief function. In this example, each of the occupied (black) and potentially occupied (gray) cells contains uncertainty where half of the mass is assigned to $\{\neg o\}$. (Figure reprinted from [Reineking and Clemens, 2013].)

directly by applying the generalized Bayesian theorem to the forward model because the forward model requires the entire map while, for the inverse model, only the marginal distribution over each cell is considered. Instead, just like the forward sensor model, the inverse model for the i -th cell is first conditioned on R (representing entirely random measurements) and C (representing the measurement cause).

$$m[x_t, z_t](Y'_i) = \sum_{R \subseteq \Theta_R} m(R) \sum_{C \subseteq \Theta_C} m[x_t, z_t, R](C) m[C, R](Y'_i) \quad (4.50)$$

The distribution of R is simply given by Eq. (4.29) and (4.30) because it is independent of the pose and the measurement. The cause C directly depends on the pose and the measurement and also on R . Finally, the cell Y'_i is independent of the pose and the measurement if both C and R are known.

The mass function $m[C, R](Y'_i)$ is categorical and expresses constraints about

possible measurement causes.

$$m[C, R](Y'_i) = 1 \quad (4.51)$$

$$Y'_i = \begin{cases} \{o\} & \text{if } R = \{\neg r\} \wedge C = \{i\}, \\ \{\neg o\} & \text{if } R = \{\neg r\} \wedge C \subseteq C_i \wedge C \neq \emptyset, \\ \Theta_Y & \text{else} \end{cases} \quad (4.52)$$

$$C_i = \{c | c \in \Theta_C, c > i\} \quad (4.53)$$

If the measurement is entirely or potentially random with $R \neq \{\neg r\}$, then the cell belief is vacuous. For the non-random case, the i -th cell must be occupied if the cause variable only contains i (i.e., there is no uncertainty regarding the cause and because it is assumed that the closest obstacle is measured, Y'_i must be occupied). Any additional cause (e.g., $C = \{i, j\}$ with $j \neq i$) would result in $Y'_i = \Theta_Y$ due of the disjunctive rule of combination. The i -th cell must be empty if the measurement cause (i.e., an occupied cell) is greater than i (i.e., farther away) and the measurement is not random. This means, all mass is assigned to $\{\neg o\}$ if the cause variable is some subset of the set C_i which represents all cells that are farther away than the i -th cell. In all remaining cases, the cell state cannot be uniquely inferred and, as a consequence, all mass is assigned to Θ_Y .

The definition of $m[C, R](Y'_i)$ together with $m(R = \{\neg r\}) = 1 - \epsilon_r$ can be used to compute the inverse model defined by Eq. (4.50).

$$m[x_t, z_t](Y'_i = \{o\}) = (1 - \epsilon_r) m[x_t, z_t, \neg r](C = \{i\}) \quad (4.54)$$

$$m[x_t, z_t](Y'_i = \{\neg o\}) = (1 - \epsilon_r) \sum_{C \subseteq C_i, C \neq \emptyset} m[x_t, z_t, \neg r](C) \quad (4.55)$$

$$\stackrel{(2.6)}{=} (1 - \epsilon_r) \text{bel}[x_t, z_t, \neg r](C_i) \quad (4.56)$$

Because the inverse model is assumed to be normalized, the mass on Θ_Y results from normalization. For the sum over R in Eq. (4.50), only the case $R = \{\neg r\}$ needs to be considered because the mass $m[C, R](Y'_i)$ on $\{o\}$ and $\{\neg o\}$ is 0 otherwise (hence the factor $1 - \epsilon_r$). For the occupied case $Y'_i = \{o\}$, C must be equal to $\{i\}$ because of Eq. (4.52). The empty case $Y'_i = \{\neg o\}$ results from summing over all subsets $C \subseteq C_i$ with $C \neq \emptyset$ where $m[C, \neg r](Y'_i = \{\neg o\}) = 1$. This sum is equal to the belief $\text{bel}[x_t, z_t, \neg r](C_i)$.

Thus, the mass $m[x_t, z_t, \neg r](C = \{i\})$ and the belief $\text{bel}[x_t, z_t, \neg r](C_i)$ need to be computed. Both can be obtained by applying the generalized Bayesian

theorem to the forward sensor model $pl[x_t, c, \neg r](z_t)$ defined by Eq. (4.36).

$$m[x_t, z_t, \neg r](C = \{i\}) \stackrel{(2.70)}{=} \eta pl[x_t, i, \neg r](z_t) \prod_{c \in \Theta_C, c \neq i} (1 - pl[x_t, c, \neg r](z_t)) \quad (4.57)$$

$$\begin{aligned} bel[x_t, z_t, \neg r](C_i) &\stackrel{(2.71)}{=} \eta \prod_{c \in \overline{C_i}} (1 - pl[x_t, c, \neg r](z_t)) \\ &\quad - \eta \prod_{c \in \Theta_C} (1 - pl[x_t, c, \neg r](z_t)) \end{aligned} \quad (4.58)$$

$$\eta^{-1} \stackrel{(2.74)}{=} 1 - \prod_{c \in \Theta_C} (1 - pl[x_t, c, \neg r](z_t)) \quad (4.59)$$

One would expect of the inverse sensor model that all cells located on the 2D arc at the measured distance receive at least some mass on $\{o\}$ because the obstacle is likely to be located somewhere on the arc. However, because these cells all roughly have the same distance to the robot, they also share very similar forward model likelihoods $pl[x_t, i, \neg r](z_t)$. Assuming the likelihood of the i -th cell is 1, then the likelihood of an equidistant cell will also be 1, which means that the product in Eq. (4.57) is 0. The reason for this behavior is that the set of all cells with likelihoods close to 1 will receive a high mass value while the singletons all receive 0 mass. This is completely justified from a theoretical standpoint because the evidence does not support any specific cell. However, it is a problem because the map belief is modeled as a combination of *marginal* distributions over single cells and, as a result, the model is incapable of representing ambiguous measurement causes.

A solution is the introduction of a prior over Θ_C which forces at least some mass to be strictly committed to the i -th cell if the cell is located at a plausible distance. Let e_i denote the “evidence” inducing this prior. The prior is defined as a simple mass function where some mass π_c is committed to $\{i\}$ while the remaining mass is assigned to the frame of discernment.

$$m[e_i](C = \{i\}) = \pi_c \quad (4.60)$$

$$m[e_i](C = \Theta_C) = 1 - \pi_c \quad (4.61)$$

The parameter π_c effectively controls how much mass will be assigned to *occupied* where a higher value for π_c implies more mass on $\{o\}$ (in the experiments presented in the next section, $\pi_c = 0.4$ is used).

The distribution of C then results from combining the prior $m_{\Theta_C}[e_i]$ with the belief $m_{\Theta_C}[x_t, z_t, \neg r]$ obtained from the forward sensor model via the generalized Bayesian theorem.

$$m_{\Theta_C}[x_t, z_t, \neg r, e_i] = m_{\Theta_C}[x_t, z_t, \neg r] \oplus m_{\Theta_C}[e_i] \quad (4.62)$$

The final mass/belief function for C resulting from the combination with the

Algorithm: Inverse sensor model	
Input: x_t, z_t	// pose, measurement
1 $L, L_i \leftarrow 1$	// recurring products
2 for $i \leftarrow 1$ to M' do	// loop over sorted cells in the cone
3 $d_i \leftarrow \sqrt{(x_{t;x} - Y'_{i;x})^2 + (x_{t;y} - Y'_{i;y})^2}$	// Euclidean cell distance
4 $pl_{z_t;i} \leftarrow \alpha \mathcal{N}(z_t; d_i, \sigma_z^2)$	// $pl[x_t, C = \{i\}, \neg r](z_t)$
5 $L \leftarrow L(1 - pl_{z_t;i})$	// product $\prod_{i=1}^{M'} (1 - pl_{z_t;i})$
6 end	
7 $\eta \leftarrow (1 - L)^{-1}$	// normalization for GBT
8 for $i \leftarrow 1$ to M' do	// second loop over cells
9 $L_i \leftarrow L_i(1 - pl_{z_t;i})$	// product $\prod_{j=1}^i (1 - pl_{z_t;j})$
10 $m_{c;i} \leftarrow \eta pl_{z_t;i} L / (1 - pl_{z_t;i})$	// $m[x_t, z_t, \neg r](C = \{i\})$
11 $bel_{c;i} \leftarrow \eta (L_i - L)$	// $bel[x_t, z_t, \neg r](C = C_i)$
12 $pl_{c;i} \leftarrow \eta pl_{z_t;i}$	// $pl[x_t, z_t, \neg r](C = \{i\})$
13 $\eta_{e_i} \leftarrow (1 - \pi_c (1 - pl_{c;i}))^{-1}$	// normalization for e_i
14 $m_{c;i;e_i} \leftarrow \eta_{e_i} (\pi_c pl_{c;i} + (1 - \pi_c) m_{c;i})$	// $m[x_t, z_t, \neg r, e_i](C = \{i\})$
15 $bel_{c;i;e_i} \leftarrow \eta_{e_i} (1 - \pi_c) bel_{c;i}$	// $bel[x_t, z_t, \neg r, e_i](C_i)$
16 $m[x_t, z_t](Y'_i = \{o\}) \leftarrow (1 - \epsilon_r) m_{c;i;e_i}$	// occupied
17 $m[x_t, z_t](Y'_i = \{\neg o\}) \leftarrow (1 - \epsilon_r) bel_{c;i;e_i}$	// empty
18 $m[x_t, z_t](Y'_i = \Theta_Y) \leftarrow 1 - m[x_t, z_t](\{o\}) - m[x_t, z_t](\{\neg o\})$	// Θ_Y
19 end	
20 return $m[x_t, z_t](Y'_1), \dots, m[x_t, z_t](Y'_{M'})$	// map belief

Figure 4.6.: Inverse sensor model algorithm for sonar. The algorithm takes the current pose x_t and measurement z_t and computes the map belief $m[x_t, z_t](Y)$. Recurring products associated with the generalized Bayesian theorem are stored in the variables L and L_i , which makes it possible to compute the entire inverse model in $O(M')$.

prior can be constructed by exploiting the fact that $m_{\Theta_C}[e_i]$ is simple.

$$m[x_t, z_t, \neg r, e_i](C = \{i\}) = \eta_{e_i} (\pi_c pl[x_t, z_t, \neg r](C = \{i\}) + (1 - \pi_c) m[x_t, z_t, \neg r](C = \{i\})) \quad (4.63)$$

$$bel[x_t, z_t, \neg r, e_i](C = C_i) = \eta_{e_i} (1 - \pi_c) bel[x_t, z_t, \neg r](C_i) \quad (4.64)$$

$$\eta_{e_i}^{-1} = 1 - \pi_c (1 - pl[x_t, z_t, \neg r](C = \{i\})) \quad (4.65)$$

Here, η_{e_i} denotes the normalization constant corresponding to the combination based on Dempster's rule in Eq. (4.62). Though computing the generalized Bayesian theorem in Eq. (4.57) and (4.58) for each cell appears to be expensive because of the products over all cells inside the measurement cone, by saving recurring products, the entire inverse model can be computed with complexity $O(M')$. An efficient algorithm based on dynamic programming for computing the inverse model is shown in Fig. 4.6.

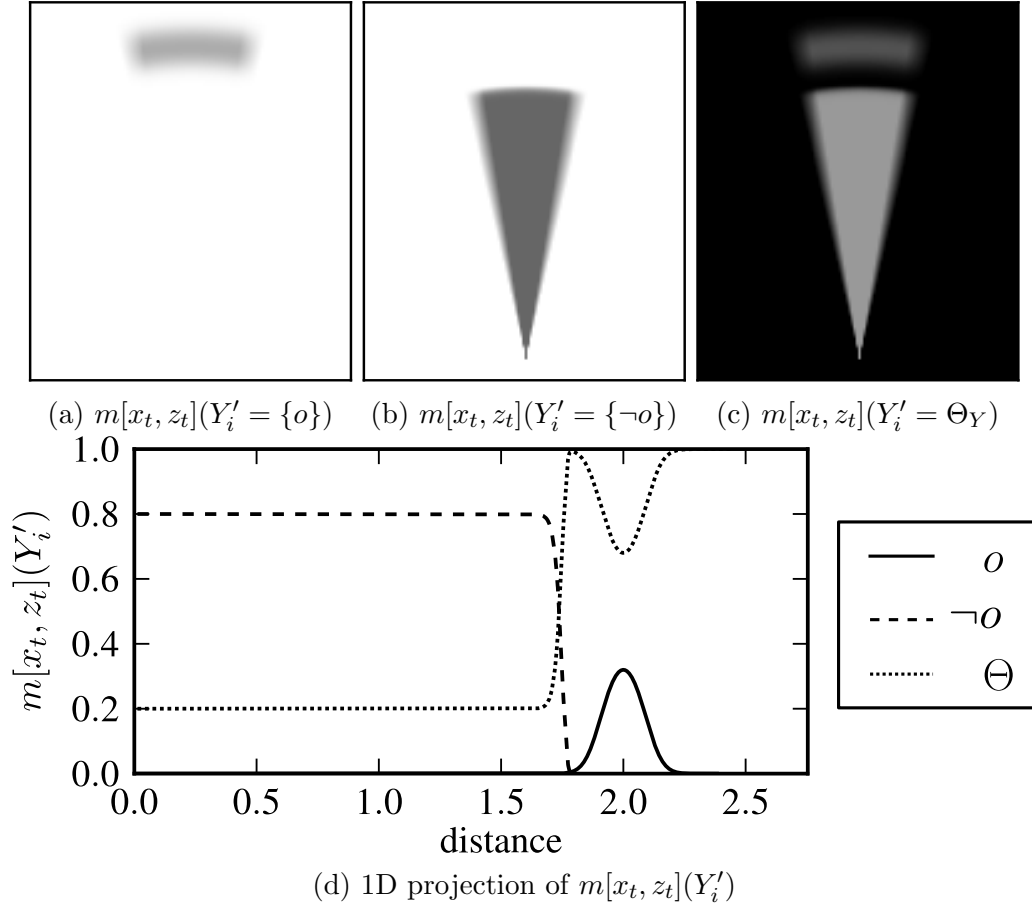


Figure 4.7.: Inverse sensor model for a measurement of $z_t = 2$. The upper three figures show the resulting 2D mass function over grid cells where darker colors indicate higher values (the cell size is very small here with 0.005 which is why the discretization is not visible). In the lower figure, a 1D projection of the 2D mass function is shown as a function of the distance from the robot. (Figure adopted from [Reineking and Clemens, 2013].)

A visualization of the inverse sensor model is shown in Fig. 4.7. For the area outside of the measurement cone, the model remains entirely agnostic as indicated by the black background in Fig. 4.7c. Fig. 4.7a shows the arc located at the measured distance where $m[x_t, z_t](Y'_i = \{o\}) > 0$. The vertical spread of the arc is the result of the Gaussian noise parameter σ_z from the forward sensor model while the magnitude is also controlled by the prior parameter π_c . As expected, the belief for *empty* is high in the area in front of the arc (see Fig. 4.7b) while the belief is vacuous in the area behind the arc (see Fig. 4.7c). The peak for Θ_Y visible in the 1D projection in Fig. 4.7d right in front of the arc is the result of the uncertainty caused by measurement noise. Note how the belief for Θ_Y is bounded below by the randomness parameter ϵ_r from the forward sensor model. Also visible are the “fuzzy” borders at the sides caused by discounting (not shown in the equations), which reflects the fact that sonar tends to be less reliable at the borders of the measurement cone.

4.5. Experimental Results

In order to demonstrate the effectiveness of the evidential FastSLAM algorithm, two experiments are conducted. In both experiments, a simulated mobile robot is steered through a virtual environment and the resulting odometry and sonar measurements are recorded. For the simulation, the *Gridmap Navigation Simulator* is used which is part of the *Mobile Robot Programming Toolkit*⁴. The robot is equipped with 8 sonar sensors, each with a 30° opening angle for the simulated sonar beam. The evidential FastSLAM algorithm is then run on the data and grid maps are constructed using different combination rules for the map update in Eq. (4.19). In particular, Dempster’s rule (Sect. 2.2.1), the conjunctive rule (Sect. 2.2.2), Yager/Dubois and Prade’s rule (which are identical for a binary frame of discernment), the cautious rule (Sect. 2.2.4), and, for comparison, Bayesian updating are used. Note that Dempster’s rule is identical to the conjunctive rule regarding localization because of the map normalization before the correction step, which is why these rules only differ with respect to the normalization of the map. For Bayesian updating, the map prior and the inverse model are pignistically transformed (the measurement plausibility remains unchanged because it is only used as an importance weight where normalization is irrelevant). Dempster’s rule is then used for incorporating the inverse model belief, which results in a Bayesian updating rule because both the prior and the pignistic inverse model are Bayesian.

4.5.1. First Experiment

In [Reineking and Clemens, 2013], the results of the first experiment were originally presented. The ground truth information for this experiment including

⁴www.mrpt.org

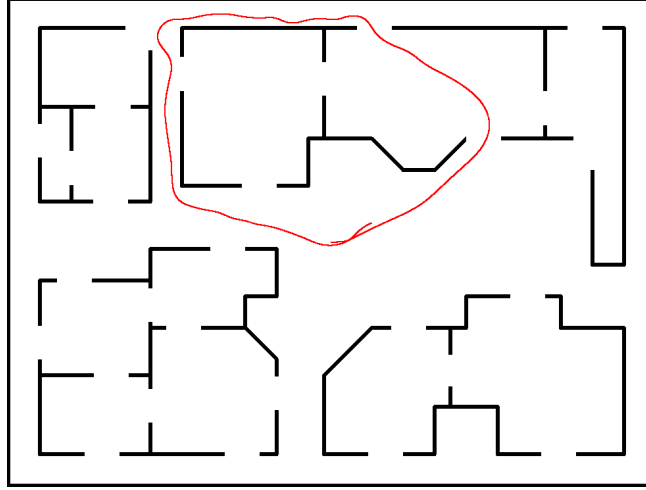


Figure 4.8.: Ground truth for the first experiment. The black lines represent walls that can be measured using sonar while the red line represents the path driven by the robot. The environment has a size of 30m×23m.

the map and the driven path are shown in Fig. 4.8. The robot drives a large loop through several rooms, which poses a particularly challenging problem for SLAM algorithms. In this experiment, 300 particles and 200×167 cells, each with a size of $15\text{cm} \times 15\text{cm}$, are used.

Fig. 4.9 shows a resulting map for each combination rule. The depicted maps are the ones with the highest product of importance weights $\prod_{t=0}^T w_t^{[k]}$ where T is the total number of time steps.⁵ These maps best fit all measurements over time and thus represent the most likely hypotheses. For each map, the mass for $\{o\}$, $\{\neg o\}$, Θ_Y , and \emptyset is shown (if the information is provided by the corresponding combination rule). In all of the shown maps, the basic layout of the environment is correctly captured by the evidential FastSLAM algorithm and only moderate distortions are visible, which are caused by the large loop where the robot has to rely almost exclusively on odometry.

Among the combination rules considered, the conjunctive rule is the only one producing mass on \emptyset and is therefore the only rule allowing for an explicit representation of conflict (though there also exists an unnormalized version of the cautious rule [Dencœux, 2008] which is not considered here). In the rightmost image in Fig. 4.9a, there is significant conflict in the vicinity of obstacles. The main reason for this conflict is measurement noise. However, the large conflicting areas visible in the bottom right of that image are caused by the accumulated localization error which results in a mismatch between older measurements (obtained at the beginning of the loop) and newer measurements (obtained at the

⁵The product is actually represented as a sum of logarithmic weights $\sum_{t=0}^T \log w_t^{[k]}$ for numerical stability because the importance weights can become quite small.

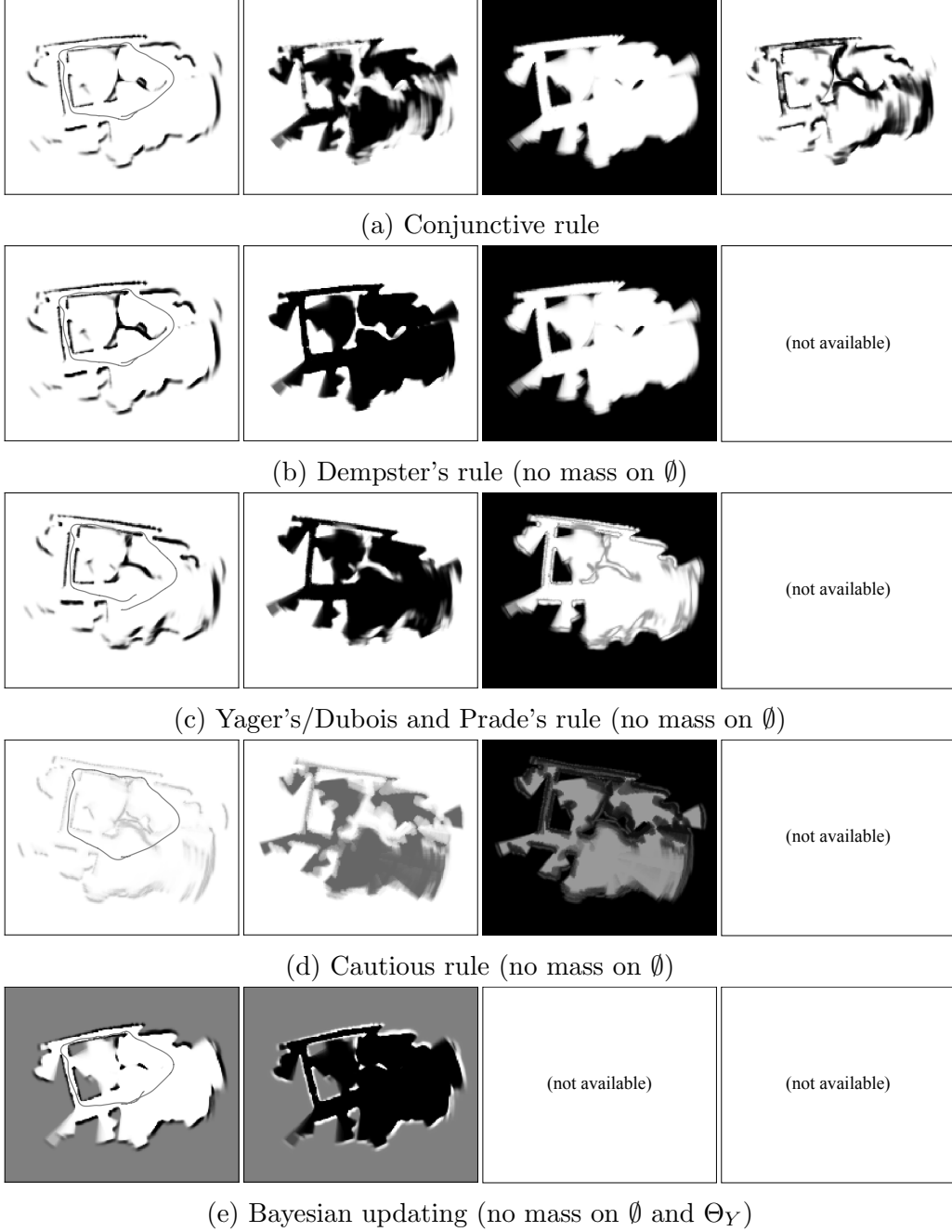


Figure 4.9.: Evidential maps generated by different combination rules in the first experiment. Each row of images corresponds to the most likely map generated by a particular combination rule. The images within each row represent the different components of the underlying belief function which are (from left to right) the mass on *occupied*, *empty*, Θ_Y , and \emptyset . Darker colors indicate higher mass values. The mass on Θ_Y represents ignorance while the mass on \emptyset represents measurement conflict. In addition, the estimated path of the robot is shown for each rule in the *occupied* maps. (Figures reprinted from [Reineking and Clemens, 2013].)

end of the loop). The conflict generally serves as a good indicator for areas that are potentially unsafe because at least some measurements indicate the possible existence of an obstacle. In contrast, in areas where the robot does not record any measurements, the mass on Θ_Y is 1, which makes the lack of information explicit.

Fig. 4.9b shows a map produced by Dempster’s rule, which is identical to the conjunctively obtained map apart from normalization.⁶ In the map produced by Yager’s/Dubois and Prade’s rule in Fig. 4.9c, all mass that would be assigned to \emptyset by the conjunctive rule is instead assigned to Θ_Y . The result is a map where more mass is assigned to Θ_Y around obstacles, expressing the ignorance of the robot regarding these cells.

The map corresponding to the cautious rule is shown in Fig. 4.9d. This rule explicitly allows for dependencies between pieces of evidence. The result is that the mapping algorithm tends to assign higher mass values to Θ_Y for conflicting measurements instead of committing to either *occupied* or *empty*. Whereas the other rules generally cause the cell belief to converge to categorical distributions after sufficiently many measurements where all mass is assigned to a single focal set, the cautious rule preserves a significant amount of uncertainty over time. This “cautious” behavior is particularly useful if there are additional information sources (e.g., additional sensors) with which the sonar-generated map is later combined.

Finally, Fig. 4.9e shows the map resulting from Bayesian updating. Some of the structures visible in the other maps are missing here. In particular compared to the map produced by the conjunctive rule, there is less information and basically no uncertainty, even for regions with missing obstacles. While some of these obstacles are not recognized as *occupied* by the conjunctive rule either, the masses on \emptyset and Θ_Y at least indicate a potential problem here.

In order to compare the quality of the estimates produced by the different combination rules, the corresponding localization error is measured. Fig. 4.10 shows the localization error over time for the most likely particle produced by each rule. All of the rules exhibit a growing error over time because of the large loop driven by the robot. Only when the robot returns to its starting position, previously mapped structures support the localization and the error decreases again. The conjunctive rule shows the lowest overall error in this experiment, which is only based on a single run though.

4.5.2. Second Experiment

The second experiment is conducted in a virtual environment consisting of a hallway and eight rooms. The ground truth map for this environment and the driven path are shown in Fig. 4.11. For this experiment, 100 particles

⁶Because of the same random seed, the maps produced by the conjunctive rule and by Dempster’s rule are *exactly* identical except for normalization.

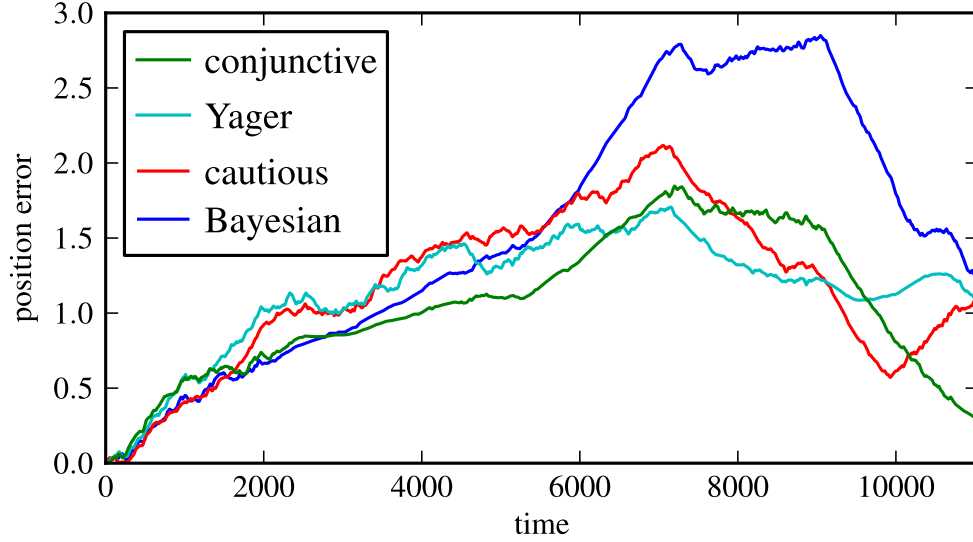


Figure 4.10.: Position error in meters over time in the first experiment resulting from different map combination rules. The error is measured as the Euclidean distance between the robot’s true position and the most-likely estimate corresponding to the highest-weighted particle. (Figure adopted from [Reineking and Clemens, 2013].)

and 267×200 cells with a size of $15\text{cm} \times 15\text{cm}$ are used. In contrast to the first experiment, the robot explores the entire environment and the path is significantly more complex. Because the robot has to return to the hallway after exploring a room, it can rely less on odometry and localize itself with respect to already mapped structures of the environment.

The resulting maps for the second experiment are shown in Fig. 4.12. Like in the first experiment, only the maps corresponding to the most likely particles are shown. Note that Dempster’s rule is omitted here because it is identical the conjunctive solution apart from normalization. While the map resulting from the conjunctive rule and Bayesian updating are mostly accurate, significant distortions are visible in the maps obtained via Yager’s rule and the cautious rule. A possible explanation is that the two latter rules tend to assign more mass to Θ_Y which means that, during localization, the evidence provided by the map tends to be weaker, thus resulting in a stronger reliance upon odometry. Though the maps shown here resulted from a single run, the experiment was repeated 10 times and the degree of distortion turned out to be very consistent for the different rules.

This can also be seen in the localization error shown in Fig. 4.13. Here, Yager’s rule and the cautious rule produce significantly larger errors while the conjunctive rule exhibits the lowest overall error. In contrast to the first experiment, the localization error is the result of averaging over 10 runs for each rule. What is noticeable is that the localization error tends to “oscillate”, which is caused by the robot exploring the different rooms. The error tends to increase

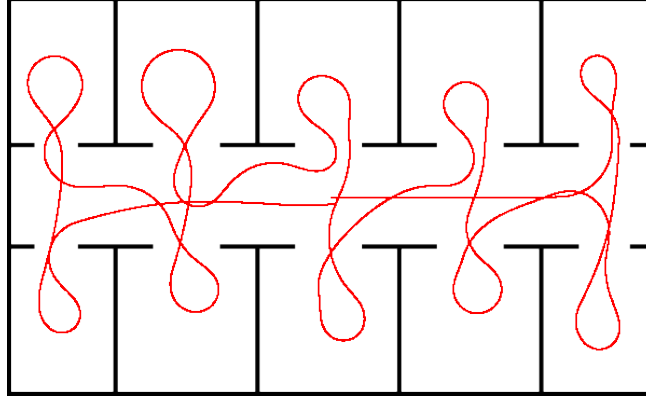


Figure 4.11.: Ground truth for the second experiment. The black lines represent walls while the red line represents the path driven by the robot. The environment has a size of $30\text{m} \times 19\text{m}$.

whenever the robot enters a room because, then, the robot can no longer reliably localize itself with respect to the hallway. Correspondingly, the error decreases again once the robot returns to the hallway. At the end of the run, the robot drives through the hallway for a longer period of time (see Fig. 4.11), which causes the error to decrease significantly for all rules.

4.6. Discussion

The evidential FastSLAM algorithm presented in this chapter uses a Rao-Blackwellized particle filter to approximate the joint distribution over path and map. Like in the original FastSLAM algorithm, the joint distribution is factorized into a path estimation problem and a mapping problem. The joint distribution is modeled as a hybrid probability/belief distribution consisting of a probabilistic path component and an evidential grid map component.

Compared to classical grid mapping, the occupancy grid maps produced by the evidential FastSLAM algorithm provide additional information about missing and conflicting evidence, the former via mass on Θ_Y and the latter via mass on \emptyset . For example, the amount of conflict can serve as a measure for potentially dangerous areas during navigating because it indicates possible model limitations (e.g., an unexpected change in the environment) and sensory ambiguity [Carlson et al., 2005]. In contrast, mass assigned to Θ_Y indicates that further measurements are required to determine the state of a cell. It also allows the robot to differentiate between the probabilistic uncertainty regarding “occupied” and “empty” and a lack of evidence. This can be useful for a robot that actively seeks to minimize uncertainty about the environment because, in a probabilistic grid map, a uniform cell distribution can mean very different things (no measurements vs. contradictory measurements) whereas an evidential grid map enables the robot to take this difference into account.

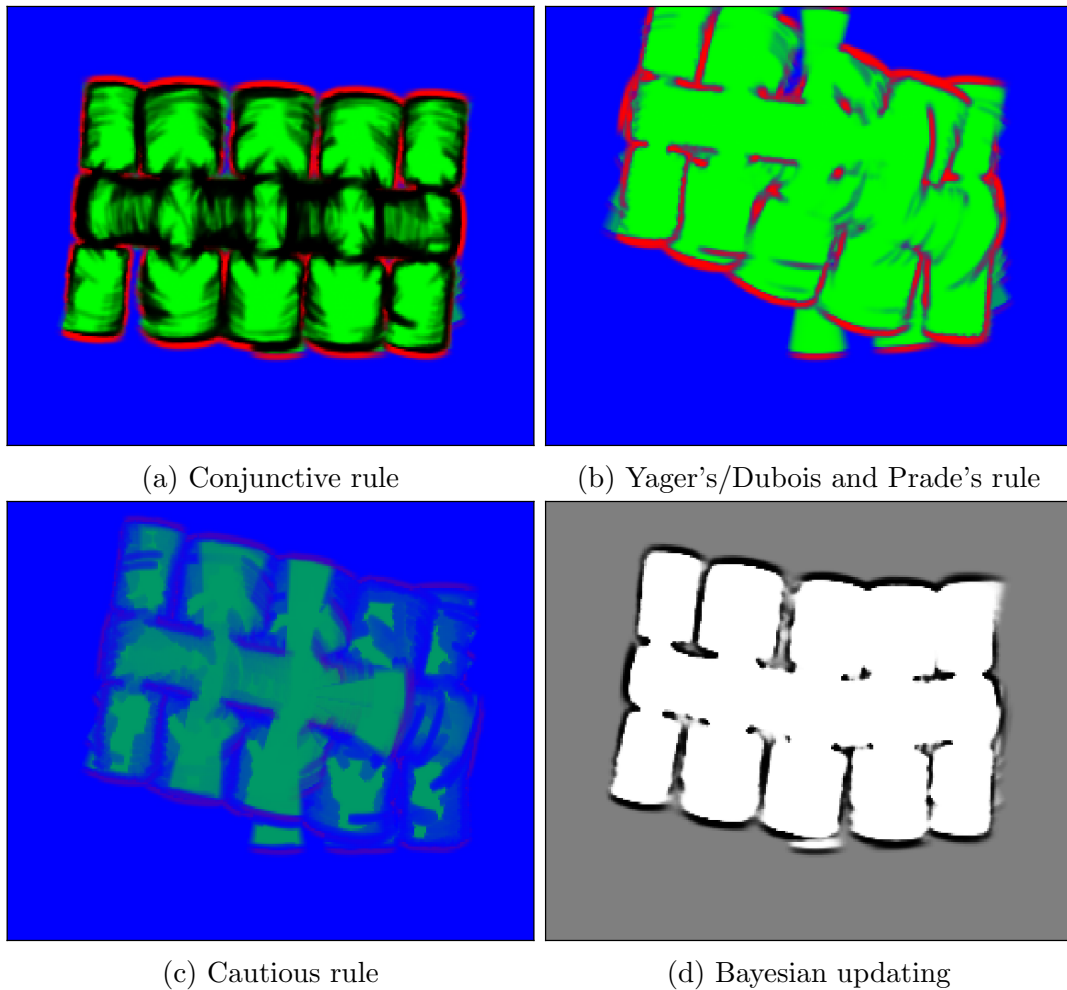


Figure 4.12.: Evidential maps generated by different combination rules in the second experiment. In map (a) to (c), the mass function for each cell is visualized in RGB color space (red represents mass on *occupied*, green represents mass on *empty*, and blue represents mass on the frame of discernment). In map (d), the Bayesian solution is shown where darker colors indicates higher occupancy probabilities.

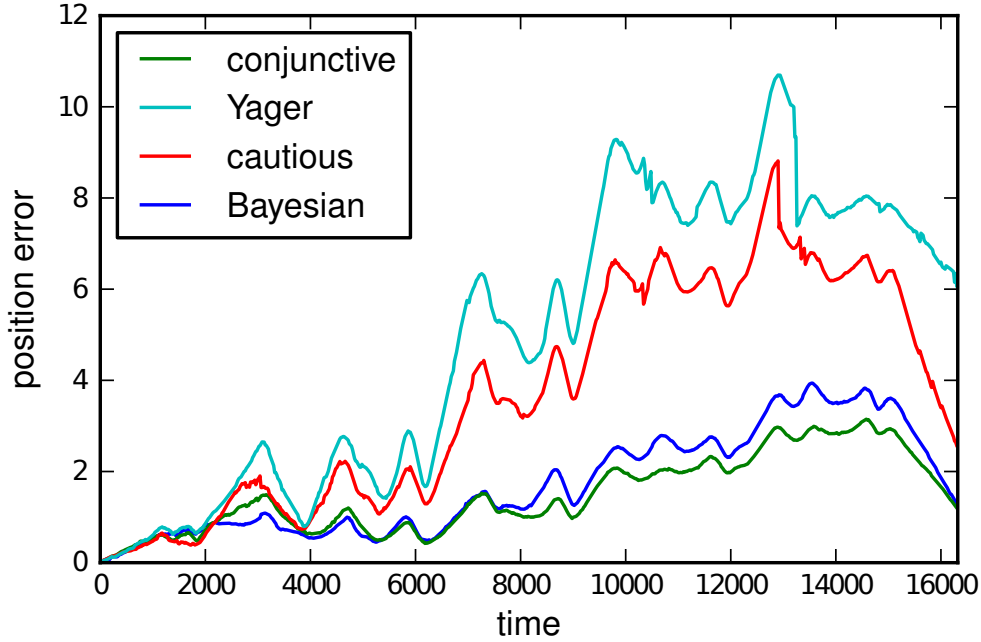


Figure 4.13.: Position error in meters over time in the second experiment resulting from different map combination rules. Like in the first experiment, the error is measured as the Euclidean distance between the robot’s true position and the estimated position belonging to the most likely particle. The error is averaged over 10 runs for each combination rule.

For possible extensions of the proposed algorithm, there are several interesting possibilities, some of which are listed here:

- Consideration of a richer, non-binary frame of discernment for the state of a cell. While the frame should not become too large because of the computational overhead, applying evidential mapping to environments where cells have multiple properties that are related in non-trivial ways could be very useful.
- Utilization of measurements in the prediction step as done in FastSLAM 2.0 [Montemerlo et al., 2003]. Because the FastSLAM approach strongly relies on the quality of the proposal distribution, this could improve the map estimate considerably.
- Learning sensor models from data. In particular for the inverse sensor model, such an approach could be beneficial because it would remove the reliance on parameters that are difficult to choose (like the prior parameter π_c). Some work regarding the construction of belief functions from data is presented in Sect. 5.4.
- Discarding the independence assumption for grid cells which could greatly

improve the map quality [Thrun, 2003]. This would require introducing some other restrictions though because handling the full belief distribution over the map space would be impossible considering the number of parameters.

- Combining different types of sensors that provide partially overlapping information. One of the strengths of the belief function framework is to fuse information sources of varying quality [Kurdej et al., 2012]. In this case, one could also use different combination rules, for example, the conjunctive rule of combination for fusing evidence of a single sensor type and the cautious rule for combining maps across different sensor types. In contrast, the probabilistic solution for mapping based on multiple sensor types is usually a simple product justified by an absolute independence assumption [Thrun et al., 2005, chapter 9].

5

Active Evidential Recognition

5.1. Introduction

In this chapter, an architecture for active evidential recognition is presented and applied to a problem of object recognition. The two main defining characteristics of the architecture are the use of belief functions for inference and an information gain strategy for selecting actions in an optimal manner. Usually, recognition is cast as a classification problem where a fixed mapping from features to classes is learned and later used to assign a class label to a given feature vector. In contrast, here, recognition is understood as a cognitive information-gathering process unfolding in time where an agent with dynamic beliefs about the world actively acquires new information. It is thus more akin to how humans acquire information about the world, i.e., not by observing and reasoning in a “one-shot” fashion but rather as an extended process of evidence collection and belief revision. The architecture presented in this chapter is inspired by the one proposed in [Schill et al., 2001], which uses bottom-up processing of sensory information and top-down reasoning based on information gain maximization to mimic the way humans actively process visual scenes by performing eye movements.

Pattern recognition problems like object recognition are inherently uncertain. There are many sources contributing to this uncertainty, ranging from sensor limitations to the ambiguous correspondence between features and classes. Thus, any algorithm which ignores this uncertainty and simply returns the most-likely class label dismisses important information.¹ The most straightforward

¹There are methods for recovering probability distributions from non-probabilistic classifiers but these methods tend to be rather ad-hoc and are not part of the original formalism.

way of formalizing a recognition problem as one of evidential inference is to treat each feature as a piece of evidence for the class. Usually, this requires conditional independence assumptions regarding the features. As a result, features can be collected over time and processed independently, thus enabling time-recursive belief updating. A particular advantage of belief functions over a probabilistic approach is that, oftentimes, it is not possible to reliably estimate the underlying probability distributions. This is the case if the number of training samples is too small compared to the number of model parameters. In this case, the problem of “overfitting” occurs, meaning that the model is only representative of the training data and fails to capture new samples. Belief functions can be used to make the lack of evidence caused by limited amounts of training data explicit and can thereby help to reduce the problem of overfitting.

When features are collected successively over time and the agent performing the collection can actively influence this process, the question becomes how the agent should select its actions such that it is optimal regarding the recognition task. An information-theoretic approach to this problem is to choose the action with the highest expected information gain, i.e., the action with the lowest expected uncertainty with respect to the class distribution after having executed the action and having collected new evidence. This raises the question of how uncertainty can be quantified when using belief functions and Sect. 2.8 describes several measures for belief function uncertainty.

In addition to the evidential inference and the information gain strategy, a third characteristic of the architecture presented in this chapter is that actions are an explicit part of the representation. In most systems, there is a clear separation between sensory information processing and motor control where the former represents the input and the latter represents the output. However, evidence from perceptual psychology and neurobiology seems to indicate that the separation of sensory and motor signals in biological systems is not strict and that motor information plays not only an important but a constituting role for perception [Noë, 2004, O’Regan and Noë, 2001, Prinz, 1990, Zetsche et al., 2009]. This is why the architecture presented here uses so-called *sensorimotor features* for the representation which combine sensory and motor information in a single vector. Further below, it is shown that this combination provides a measurable advantage over purely sensory features in terms of recognition accuracy.

The remainder of this chapter is structured as follows. Work related to classification based on belief functions is briefly discussed in Sect. 5.2. In addition, the origin of the recognition architecture and its applications to other domains are described. The architecture itself including the evidential inference and the information gain strategy are presented in Sect. 5.3. The problem of constructing models from limited amounts of training data using belief functions is described in Sect. 5.4 where several approaches are compared. In Sect. 5.5, the recognition architecture, along with the different model learning approaches, is applied to an object recognition problem. The chapter concludes with a discussion of the presented architecture in Sect. 5.6.

5.2. Related Work

For classification within the belief function framework, two general approaches can be distinguished. The first is a discriminative one where several pieces of evidence are represented as belief functions and then combined using an appropriate combination rule [Quost et al., 2011]. In the simplest case, this evidence can be the distance between the feature vector of a new sample and a training sample, resulting in an evidential nearest neighbor classification [Dencœux, 1995, Dencœux, 2000]. The evidence can also result from the application of non-evidential classifiers where belief functions are used to perform classifier fusion [Kuncheva et al., 2001, Hady et al., 2011].

The other approach to classification in the belief function framework is a generative one. Here, the generalized Bayesian theorem is applied and the posterior class distribution is computed from a prior and class-conditional distributions over the features [Smets, 1998a]. In fact, both the discriminative and generative approach can be derived using the generalized Bayesian theorem and both are identical for certain distributions [Dencœux and Smets, 2006].² In principle, the generative approach is more powerful because it models the joint distribution of classes and features. However, in order to cope with the high dimensionality of the joint feature space, a conditional independence assumption between features is usually made. A nice property of the generative approach is that each class has its own model, making it very easy to add additional classes without modifying previously constructed models.

The information gain maximization principle in conjunction with evidential inference was originally proposed in a series of papers [Schill, 1997, Schill, 1995, Schill et al., 1991]. In these papers, the expected information gain of an action is defined as the L_1 distance between the current belief and a predicted belief after having performed an action (the underlying mass functions are expressed as vectors for the distance computation). In order to reduce computational complexity, the hypothesis space is restricted to a tree structure where only certain focal sets are allowed [Gordon and Shortliffe, 1985]. The resulting architecture based on evidential inference and information gain maximization was applied in domains such as object recognition [Schill et al., 2001], self-localization [Zetsche et al., 2008] (which is extended by a temporal inference mechanism in [Reineking et al., 2010, Zetsche et al., 2008], see Chap. 3), and scene categorization in conjunction with a domain ontology [Schill et al., 2009].

With respect to the architecture presented in this chapter, there are some key differences:

- The inference in the cited works is based on a discriminative model where features are assumed to provide distinct pieces of evidence, which are then combined using Dempster’s rule. In contrast, the inference presented

²This is very similar to discriminative-generative model pairs in the Bayesian framework [Sutton and McCallum, 2010].

in the next section uses a generative model and is based on the generalized Bayesian theorem. This generative version was first proposed in [Reineking et al., 2010].

- Instead of restricting the hypothesis space to a tree structure for reducing computational complexity, the full hypothesis space is used and complexity is reduced by means of a Monte-Carlo algorithm. This is necessary because the disjunctive rule of combination used in the generative inference approach produces focal sets that generally cannot be expressed in a tree-structured hypothesis space.
- The expected information gain is defined as the difference between the current uncertainty (quantified by an evidential uncertainty measure) and the expected uncertainty after having executed an action where the pignistic transformation is used to compute the expected value.
- In some of the cited works, there is an additional bottom-up component where, in the case of vision, a saliency detector is used to determine possible actions based on “interesting” image regions [Zetzsche et al., 1993]. Here, the set of possible actions is defined in advance.

A key question in all of the cited applications is how the underlying belief functions are constructed. One possibility is to rely on expert knowledge (e.g., represented as a domain ontology), though this is usually problematic because such knowledge tends to be very “high-level” and hard to capture quantitatively. This is why, in [Reineking et al., 2011] and [Reineking et al., 2009], a hybrid approach is proposed that combines a domain ontology with a statistical model for the problem of scene categorization. The statistical model describes relations between scene categories and contained objects and it is derived from co-occurrence statistics in the image database LabelMe [Russell et al., 2008]. In contrast, the ontological model provides high-level constraints about scenes and objects (e.g., “*kitchens contain means for cooking*”). Both models are combined using belief functions because belief functions are capable of capturing the probabilistic information of the statistical model as well as the set-based propositions resulting from the domain ontology.

In this chapter, an alternative approach for constructing belief functions from data in a fully automated fashion without relying on expert knowledge is investigated. Because the true probability distribution underlying the data can only be approximated based on a limited number of training samples, belief functions are used to make the lack of evidence resulting from the limited sample count explicit (see Sect. 5.4).

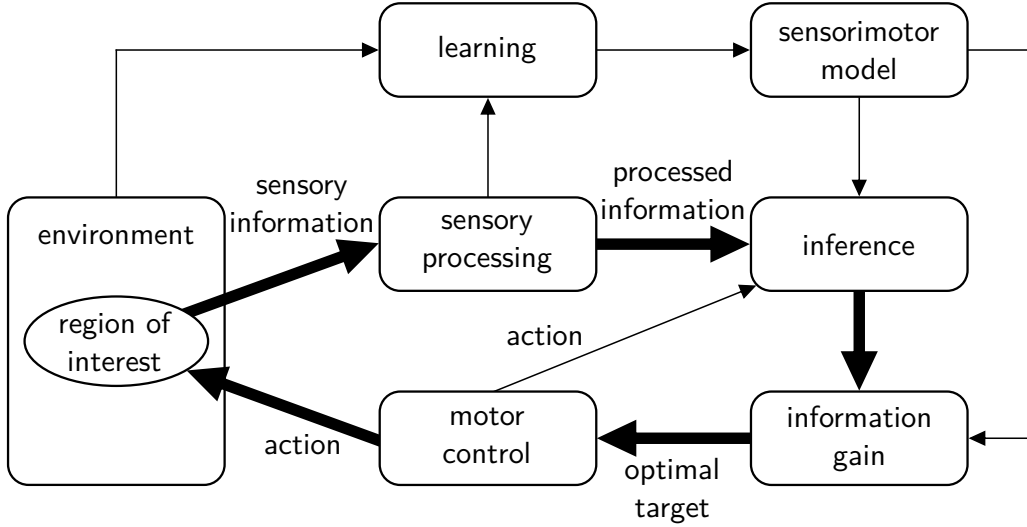


Figure 5.1.: Architecture for active evidential recognition. The continuous evidence selection and update cycle is indicated by the thick arrows. Here, sensory information from the environment is actively selected based on the current belief (computed by the inference module) and the information gain strategy. The newly collected evidence is then used to update the current belief which, in turn, leads to another information-gathering action.

5.3. Recognition Architecture

A schematic overview of the recognition architecture is depicted in Fig. 5.1. The architecture is based on a continuous cycle of collecting the “most informative” features in the environment and using these features to update the current belief distribution. Once an information-gathering action has been performed, features are first extracted from the newly collected information. The combination of the executed action and the processed sensory information forms a sensorimotor feature which is passed to the inference module. The inference module maintains a belief distribution over time and updates this distribution by incorporating new evidence based on a previously-learned sensorimotor model (see Sect. 5.4). The updated belief distribution is then passed to the information gain module where the optimal next action is determined by maximizing the expected information gain with respect to the current belief distribution using the sensorimotor model in order to predict the effect of an action. The action with the highest expected information gain is then executed by the motor control module leading to the collection of a new piece of evidence, after which the cycle starts over. Note that feature extraction, motor control, and the sensorimotor model are mostly domain-specific. In contrast, inference and information gain computation are very generic and can be directly applied to other domains.

The belief network underlying the inference is shown in Fig. 5.2. Let $X \subseteq \Theta$

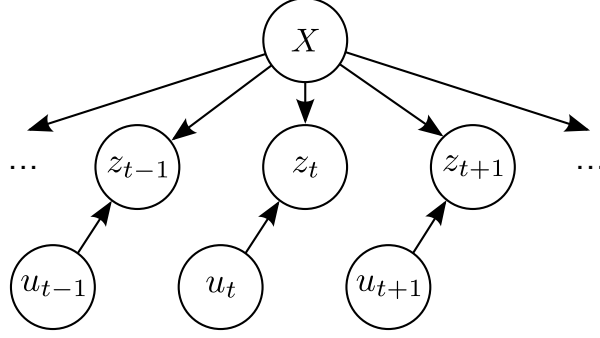


Figure 5.2.: Belief network for classification. Each feature z_t collected at time t is assumed to depend only on the class X and the action u_t preceding its collection. Each action u_t is assumed to be independent of the class.

be an evidential variable representing the class and let $\Theta = \{x_1, x_2, \dots\}$ be the corresponding finite frame of discernment where each x_i denotes a class. Furthermore, let $z_{1:t}$ denote the sequence of sensory features collected over time and let $u_{1:t}$ denote the sequence of performed actions. Each tuple (z_t, u_t) forms a sensorimotor feature. The corresponding frames of discernment $z_t \in \Theta_z$ and $u_t \in \Theta_u$ are finite and time-invariant. Sensorimotor features are assumed to be conditionally independent given the class. This resembles a Naive Bayes model, although more general models could be considered (e.g., with dependencies between subsequent features, see [Kluth et al., 2013] for an investigation of sensorimotor models with fewer independence assumptions). If there is prior evidence e_0 about the class, the architecture can incorporate this as well. The recognition problem to be solved by the architecture is the computation of the class belief distribution $m_\Theta[z_{1:t}, u_{1:t}, e_0]$ given all evidence.

The next two sections present solutions to the problems of inference and information gain maximization.

5.3.1. Inference

Because of the assumption of conditional independence of features, the final belief distribution can be decomposed into a series of distributions, each induced by a particular piece of evidence. This series of belief functions can be combined using Dempster's rule due to conditional non-interactivity (all belief functions are assumed to be normalized).

$$m_\Theta[z_{1:t}, u_{1:t}, e_0] \stackrel{(2.67)}{=} m_\Theta[e_0] \oplus \left(\bigoplus_{i=1}^t m_\Theta[z_i, u_i] \right) \quad (5.1)$$

In case there is no prior evidence e_0 , the corresponding belief function $m_\Theta[e_0]$ is vacuous.

Using this decomposition, the update can be performed recursively over time where the belief at time $t - 1$ is used to compute the belief at time t .

$$m_{\Theta}[z_{1:t}, u_{1:t}, e_0] = m_{\Theta}[z_{1:t-1}, u_{1:t-1}, e_0] \oplus m_{\Theta}[z_t, u_t] \quad (5.2)$$

The belief $m_{\Theta}[z_t, u_t]$ induced by sensorimotor feature (z_t, u_t) can be computed from the likelihoods $pl[x](z_t, u_t)$ using the generalized Bayesian theorem.

$$m_{\Theta}[z_t, u_t](X) \stackrel{(2.70)}{=} \eta \prod_{x \in X} pl[x](z_t, u_t) \prod_{x \in \bar{X}} (1 - pl[x](z_t, u_t)) \quad (5.3)$$

By assuming the a priori belief for u_t is vacuous, each likelihood $pl[x](z_t, u_t)$ can be expressed as a marginal distribution over z_t by using the definition of conditional plausibility.

$$pl[x](z_t, u_t) \stackrel{(2.57)}{=} pl[x, u_t](z_t) \quad (5.4)$$

The model which provides the basis for inference and which has to be learned thus consists of two distributions: the prior $m_{\Theta}[e_0]$ and the sensorimotor likelihood $pl[x, u_t](z_t)$. The process of constructing these distributions from data is described in Sect. 5.4.

In practice, solving the above equations exactly is infeasible because the frames of discernment are usually too large (e.g., in the application in Sect. 5.5, $|\Theta| = 10$ and $|\Theta_z| = 100$). For this reason, an approximate solution is computed. A Monte-Carlo algorithm for updating the belief distribution in an efficient manner using importance sampling is shown in Fig. 5.3. It is essentially the same algorithm used in the correction step of the particle filter presented in Sect. 3.4.2. The only difference is that the belief is not explicitly represented as a particle set. Instead, samples are drawn on-the-fly from the prior distribution $m_{\Theta}[e_0, z_{1:t-1}, u_{1:t-1}]$ (line 3). In order to reduce variance, this sampling process is based on quantizing the mass values (not shown in Fig. 5.3). As a result, resampling is omitted and the importance weights $w_t^{[k]}$ are directly added to the updated mass function (line 16).

Using this algorithm, the belief update can be performed with linear time complexity $O(K |\Theta|)$ where K denotes the number of samples drawn from the prior distribution (e.g., $K = 10,000$ in Sect. 5.5). Finally, in order to perform classification, the pignistic transformation is applied to the inferred mass distribution and the singleton with the highest pignistic probability is returned.

5.3.2. Information Gain

As described above, the architecture selects the action with the highest expected information gain. Let U denote an uncertainty measure for belief functions (see Sect. 2.8). The expected information gain $I(u_t)$ of action u_t is defined as the expected reduction in uncertainty after having performed action u_t

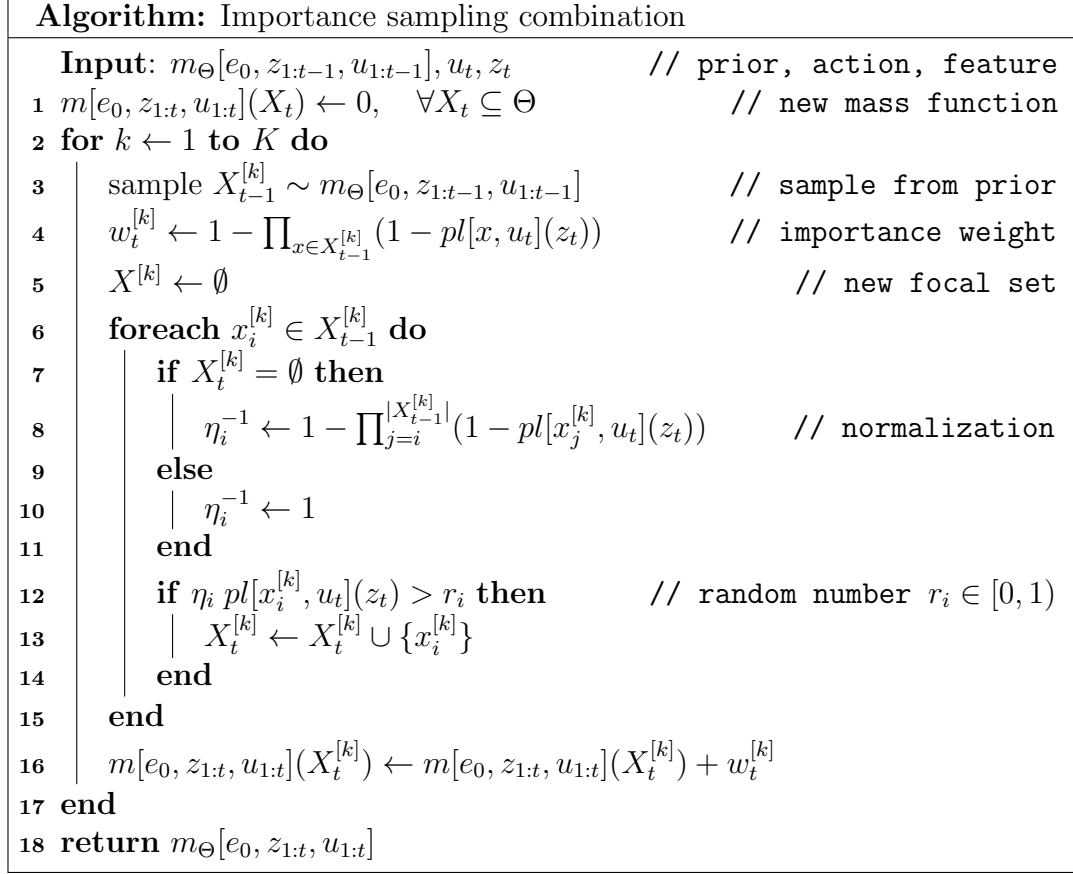


Figure 5.3.: Importance sampling algorithm for approximating the belief update based on a newly observed feature z_t , the corresponding action u_t , and the prior belief $m_{\Theta}[e_0, z_{1:t-1}, u_{1:t-1}]$. The entire algorithm resembles the correction step of the particle filter in Sect. 3.4.2 except for the absence of a resampling step. In particular, lines 5–15 are equivalent to the function `compatible_sample` shown in Fig. 3.6.

[Thrun et al., 2005, chapter 17].

$$I(u_t) = U(m_{\Theta}[z_{1:t-1}, u_{1:t-1}]) - E_{z_t}(U(m_{\Theta}[z_{1:t}, u_{1:t}])) \quad (5.5)$$

Because the new measurement z_t corresponding to action z_t is not known, the expected value of the resulting uncertainty $U(m_{\Theta}[z_{1:t}, u_{1:t}])$ with respect to z_t has to be considered. Throughout this chapter, the uncertainty measure U is defined as the pignistic entropy H_{BetP} introduced in Eq. (2.89).³ The pignistic entropy is used here because it can be computed efficiently and because the most-likely class is determined based on the pignistic transformation, meaning that the

³The uncertainty measure can be freely chosen because the information gain algorithm does not require it to satisfy any particular properties (i.e., should at some point a commonly accepted uncertainty measure for belief functions emerge, it could be directly used).

pignistic entropy accurately describes the probabilistic uncertainty associated with the classification decision. Note that if the distributions of X and z_t are Bayesian, U would reduce to the Shannon entropy and the expected information gain would be equal to the mutual information of class X and sensorimotor feature (Z_t, u_t) [Palm, 2012, chapter 3].

Finding the action u_t^* associated with the highest expected information gain simply requires computing the gain for every possible action u_t and choosing the action that maximizes $I(u_t)$. Let $\Theta_{u;t} \subseteq \Theta_u$ denote the set of possible actions remaining at time t . Because performing the same action twice does not make any sense assuming the world is otherwise static, $\Theta_{u;t}$ only contains those actions that have not yet been performed. Maximizing the expected information gain $I(u_t)$ is equivalent to minimizing the expected uncertainty after having executed action u_t because the current uncertainty is constant (it is independent of u_t).

$$u_t^* = \arg \max_{u_t \in \Theta_{u;t}} I(u_t) \quad (5.6)$$

$$\stackrel{(5.5)}{=} \arg \max_{u_t \in \Theta_{u;t}} \underbrace{U(m_\Theta[z_{1:t-1}, u_{1:t-1}])}_{\text{const.}} - E_{z_t}(U(m_\Theta[z_{1:t}, u_{1:t}])) \quad (5.7)$$

$$= \arg \min_{u_t \in \Theta_{u;t}} E_{z_t}(U(m_\Theta[z_{1:t}, u_{1:t}])) \quad (5.8)$$

Computing this expected value requires constructing the pignistic transformation of the distribution over z_t as described in Sect. 2.7. The pignistic transformation is computed from the corresponding mass function $m_{\Theta_z}[z_{1:t-1}, u_{1:t}]$.

$$E_{z_t}(U(m_\Theta[z_{1:t}, u_{1:t}])) \stackrel{(2.84)}{=} \sum_{z_t \in \Theta_z} U(m_\Theta[z_{1:t}, u_{1:t}]) \text{BetP}[z_{1:t-1}, u_{1:t}](z_t) \quad (5.9)$$

$$\text{BetP}[z_{1:t-1}, u_{1:t}](z_t) \stackrel{(2.83)}{=} \sum_{Z_t \subseteq \Theta_z, z_t \in Z_t} \frac{m[z_{1:t-1}, u_{1:t}](Z_t)}{|Z_t|}, \quad \forall z_t \in \Theta_z \quad (5.10)$$

The uncertainty $U(m_\Theta[z_{1:t}, u_{1:t}])$ after having observed z_t is computed using the inference algorithm shown in Fig. 5.3 and then applying the uncertainty measure U . The mass function $m_{\Theta_z}[z_{1:t-1}, u_{1:t}]$ can be obtained by conditioning on class X . By exploiting independence for feature Z_t (it only depends on class X and action u_t) and for class X (action u_t without the corresponding feature z_t does not influence the class belief), the resulting expressions can be simplified.

$$m_{\Theta_z}[z_{1:t-1}, u_{1:t}] \stackrel{(2.63)}{=} \sum_{X \subseteq \Theta} m_{\Theta_z}[X, z_{1:t-1}, u_{1:t}] m[z_{1:t-1}, u_{1:t}](X) \quad (5.11)$$

$$= \sum_{X \subseteq \Theta} m_{\Theta_z}[X, u_t] m[z_{1:t-1}, u_{1:t-1}](X) \quad (5.12)$$

The mass function $m_\Theta[z_{1:t-1}, u_{1:t-1}]$ simply represents the belief at time $t-1$ and is thus directly available. Because the feature distribution $m_{\Theta_z}[X, u_t]$ is

conditioned on a set of classes X , the disjunctive rule of combination can be applied in order to construct it from the singleton-conditioned distributions $m_{\Theta_z}[x_i, u_t]$ with $x_i \in X$.

$$m[X, u_t](Z_t) \stackrel{(2.60)}{=} \sum_{\left(\bigcup_{i: x_i \in X} Z_{t;i}\right)=Z_t} \prod_{x_i \in X} m[x_i, u_t](Z_{t;i}) \quad (5.13)$$

Complexity Reduction

In principle, the above equations can be used to compute the expected information gain. However, for most realistic scenarios, exactly solving these equations is infeasible. The time complexity of computing the pignistic probability distribution in Eq. (5.10) is at least $O(2^{|\Theta_z|+|\Theta|})$ and thus much higher than for inference. While the inference can be effectively approximated using sampling, applying such an approach to the computation of the pignistic probability would be problematic because it would require drawing samples from the joint space $\Theta \times \Theta_z$.

In order to simplify the problem, it is assumed that there exists a Bayesian prior for the feature z_t . Unless there is evidence indicating otherwise, this prior $P(z_t)$ is assumed to be uniform. The introduction of the prior serves no other purpose than to make computing the expected information gain feasible. In practice, this prior seems to have no negative effect on the behavior of the system, though it would be advisable to investigate the effects in more detail in the future (e.g., by comparing the solution to a Monte-Carlo approach where a sufficiently large number of samples is drawn from the joint space $\Theta \times \Theta_z$).

Let e_z denote the “evidence” representing the uniform Bayesian prior over z_t . The mass function $m_{\Theta_z}[z_{1:t-1}, u_{1:t}]$ underlying the pignistic probability distribution in Eq. (5.10) is combined with the prior induced by e_z using Dempster’s rule. A combination with a Bayesian belief function corresponds to a probability-plausibility product (see proof in Sect. A.1). Because the prior is assumed to be uniform, the pignistic probability distribution $BetP[z_{1:t-1}, u_{1:t}, e_z](z_t)$ is simply proportional to the plausibility $pl[z_{1:t-1}, u_{1:t}](z_t)$ where normalization can be trivially performed afterwards.

$$m[z_{1:t-1}, u_{1:t}, e_z](Z_t) \stackrel{(A.1)}{\propto} \begin{cases} P(z_t) pl[z_{1:t-1}, u_{1:t}](z_t) & \text{if } Z_t = \{z_t\} \\ 0 & \text{else} \end{cases} \quad (5.14)$$

$$BetP[z_{1:t-1}, u_{1:t}, e_z](z_t) \propto P(z_t) pl[z_{1:t-1}, u_{1:t}](z_t) \quad (5.15)$$

Like before, the plausibility $pl[z_{1:t-1}, u_{1:t}](z_t)$ is conditioned on the class X and by exploiting the various independence properties, the resulting equation

can be simplified.

$$pl[z_{1:t-1}, u_{1:t}](z_t) \stackrel{(2.63)}{=} \sum_{X \subseteq \Theta} pl[X, z_{1:t-1}, u_{1:t}](z_t) m[z_{1:t-1}, u_{1:t}](X) \quad (5.16)$$

$$= \sum_{X \subseteq \Theta} pl[X, u_t](z_t) m[z_{1:t-1}, u_{1:t-1}](X) \quad (5.17)$$

Finally, the disjunctive rule of combination is used to compute the plausibility $pl[X, u_t](z_t)$ from the singleton-conditioned plausibilities $pl[x, u_t](z_t)$. These plausibilities represent the previously-learned sensorimotor model and they are the same like the ones used for inference in Eq. (5.3).

$$pl[X, u_t](z_t) \stackrel{(2.62)}{=} 1 - \prod_{x \in X} (1 - pl[x, u_t](z_t)) \quad (5.18)$$

The sum over all subsets $X \subseteq \Theta$ in Eq. (5.17) can be approximated by drawing K samples from the prior distribution $m_{\Theta}[z_{1:t-1}, u_{1:t-1}]$. As a result, the complexity of computing the pignistic probability over z_t is reduced to $O(K |\Theta|)$. Combined with the iteration over all actions in Eq. (5.6) and the sum over all features in Eq. (5.9), the overall time complexity of maximizing the expected information gain is $O(K |\Theta_u| |\Theta_z| |\Theta|)$. The corresponding algorithm for computing the expected information gain of an action u_t is shown in Fig. 5.4. Note that the number of samples K used here can be different than the number of samples used for the actual inference (usually a smaller K can be used for the information gain because the consequences of approximation errors are not as severe as in case of the inference). Depending on the sizes of the involved frames, the computational expense can be quite significant and, with respect to the concrete scenario, one has to consider the trade-off between the computational costs and the costs of choosing suboptimal actions.

5.4. Model Learning

For a long time, the focus in belief function theory research has been on the problem of combining evidence. In comparison, the question of where belief functions come from in the first place received little attention. Oftentimes, the construction of belief functions is rather ad-hoc or it requires the availability of “expert knowledge”. Therefore, it would be useful to have more principled methods for deriving belief functions directly from data.

Before turning to the question of *how* belief functions can be constructed from data, the question of *why* one would want to model data with belief functions has to be addressed. Even with data collection becoming cheaper in many domains, the problems one might want to solve using such data are becoming more complex as well. For example, the plausibility distribution $pl[x, u_t](z_t)$

Algorithm: Expected information gain	
Input: $m_{\Theta}[e_0, z_{1:t-1}, u_{1:t-1}], u_t$	// belief at $t-1$, action
1 $U_{t-1} \leftarrow U(m_{\Theta}[e_0, z_{1:t-1}, u_{1:t-1}])$	// current uncertainty
2 for $i \leftarrow 1$ to $ \Theta_z $ do	// compute probability of z_t
3 $p_i \leftarrow 0$	// initialize $BetP[z_{1:t-1}, u_{1:t}, e_z](z_{t,i})$
4 for $k = 1$ to K do	
5 sample $X_{t-1}^{[k]} \sim m_{\Theta}[e_0, z_{1:t-1}, u_{1:t-1}]$	// sample from prior
6 $pl_{i,k} \leftarrow 1 - \prod_{x \in X_{t-1}^{[k]}} (1 - pl[x, u_t](z_{t,i}))$	// $pl[X_{t-1}^{[k]}, u_t](z_{t,i})$
7 $p_i \leftarrow p_i + pl_{i,k}$	// update probability
8 end	
9 end	
10 $p_{\Sigma} \leftarrow \sum_{i=1}^{ \Theta_z } p_i$	
11 $p_i \leftarrow p_i p_{\Sigma}^{-1}, \quad 1 \leq i \leq \Theta_z $	// normalization
12 $E(U_t) \leftarrow 0$	// initialize expected uncertainty
13 for $i \leftarrow 1$ to $ \Theta_z $ do	// compute expected uncertainty
14 $m_{z_{t,i}, u_t} \leftarrow m_{\Theta}[e_0, z_{1:t-1}, z_{t,i}, u_{1:t}]$	// inference using $z_{t,i}, u_t$
15 $U_{t,i} \leftarrow U(m_{z_{t,i}, u_t})$	// uncertainty for $z_{t,i}, u_t$
16 $E(U_t) \leftarrow E(U_t) + p_i U_{t,i}$	// update expected uncertainty
17 end	
18 return $U_{t-1} - E(U_t)$	// expected information gain

Figure 5.4.: Algorithm for computing the expected information gain of action u_t given the belief $m_{\Theta}[e_0, z_{1:t-1}, u_{1:t-1}]$ at time $t-1$. In the first part (lines 2–11), the pignistic probability distribution $BetP[z_{1:t-1}, u_{1:t}, e_z](z_t)$ is computed by drawing samples from the prior distribution and updating the probabilities using the model $pl[x, u_t](z_t)$. Because the pignistic probability values are only proportional to the corresponding plausibility values, normalization has to be performed (lines 10–11). In the second part (lines 12–17), the expected uncertainty $E_{z_t}(U(m_{\Theta}[z_{1:t}, u_{1:t}]))$ is computed using the approximated pignistic probability distribution and the inference algorithm shown in Fig. 5.3.

introduced in the previous section and described in more detail in the next section depends on three variables with 50,000 possible joint assignments meaning that 50,000 parameters have to be estimated. The problem is that the number of parameters grows exponentially with the number of variables without additional constraints (e.g., allowing dependencies between feature z_t and the previous feature z_{t-1} already results in 5,000,000 parameters). As a result, estimating a probability distribution over such a joint space is difficult because multiple observations are needed for each parameter combination in order to obtain a reliable probability estimate.

In contrast, belief functions are capable of expressing missing information by assigning mass values to arbitrary subsets of the frame of discernment. For example, in the extreme case where no samples are available during training, the evidence (or lack thereof) can still be accurately described by a vacuous belief function. For classification, the problem of not having a sufficient number of training samples is very common and it leads to overfitting where the model correctly describes the collected samples but not the true distribution of samples. In recent years, several approaches for constructing belief functions from data have been proposed where the problem of small sample counts is explicitly addressed [Dencœux, 2006, Aregui and Dencœux, 2008, Walley, 1996] (see also [Ferson et al., 2003, Szczot et al., 2012]). These approaches are described and compared in this section and then applied to an object recognition problem in the next section.

In all of the presented approaches, the aim is to find a belief function describing a discrete random variable X defined over a finite domain Θ with $d = |\Theta|$.⁴ The random variable is assumed to be distributed according to some unknown categorical⁵ probability distribution $P^* \in \mathbb{P}$ where \mathbb{P} denotes the set of all categorical probability distributions defined over Θ . All that is available for estimating the parameters of P^* is an independent and identically distributed random sample drawn from P^* .

Let n_i denote the absolute frequency of event $x_i \in \Theta$ in the random sample and let $\mathbf{n} = (n_1, \dots, n_d)$ denote the vector of all observed frequencies. The total size of the random sample is then $n = \sum_{i=1}^d n_i$. If the sample size n is sufficiently large compared to the domain size d , there is no problem and one could simply accept the relative frequency n_i/n of each x_i as a good approximation for the true probability $P^*(x_i)$. In contrast, if n is not large enough, a belief function can make this lack of evidence explicit. Intuitively, the smaller the sample size, the less committed the resulting belief function should be.

Three different approaches for constructing a belief function from a random sample are presented in this section. They are all based on the idea of deriving lower and upper probability bounds from which a normalized belief function can

⁴An approach for real-valued random variables is described in [Aregui and Dencœux, 2008], however, the focus here is only on discrete random variables.

⁵Not to be confused with “categorical belief functions”, see Sect. 2.1.5.

then be obtained. For comparison, two probabilistic approaches are presented as well. Approaches where the parameters of the unknown distribution are modeled as random variables are not considered here because it would significantly increase the computational complexity of the resulting inference process and it would make comparisons more difficult. Note that all of the approaches presented here are implemented in the *PyDS* library described in appendix B. For a more detailed exposition of the different approaches, the reader is encouraged to refer to the corresponding articles.

5.4.1. Maximum Likelihood

As described above, the simplest solution to constructing a probability function P based on an observed random sample is to accept the relative frequencies as probabilities.

$$P(x_i|\mathbf{n}) = \frac{n_i}{n}, \quad 1 \leq i \leq d \quad (5.19)$$

This corresponds to a *maximum likelihood* estimate and it is the optimal solution for $n \rightarrow \infty$. However, if n is small, this solution is problematic, in particular for cases like $n_i = 0$, because x_i would be completely rejected during inference while the true probability could be non-zero.

5.4.2. Laplace Smoothing

In order to avoid the problem of assigning probability 0 to an event simply because it does not appear in the random sample, a common practice is to add a small value $s > 0$ to each count n_i . This (probabilistic) approach is called *Laplace smoothing* or *additive smoothing*.

$$P(x_i|\mathbf{n}) = \frac{n_i + s}{n + s d}, \quad 1 \leq i \leq d \quad (5.20)$$

A typical parameter choice is $s = 1$, which is also used for the application in Sect. 5.5.

5.4.3. Imprecise Dirichlet Model

The approach proposed in [Walley, 1996] is called the *Imprecise Dirichlet Model* and it is conceptually similar to Laplace smoothing. It was originally proposed in the context of the imprecise probability framework [Walley, 2000] but the solution can also be expressed as a belief function. The author considers the problem of drawing marbles with different colors from a bag and raises the question of how one can construct a sample space without prior knowledge about possible colors. As a consequence, the author proposes a “representation invariance principle” (RIP) which states that the belief values should not depend on the choice of the sample space.

The model assumes “imprecise” Dirichlet priors for the parameters of the unknown distribution P^* , meaning that the parameters of the Dirichlet priors are themselves unknown. This induces a lower bound P^- and an upper bound P^+ for the probability function P^* . Generally, bounds for a small value of n should be large while they should be narrow for a large value of n .

$$P^-(x_i|\mathbf{n}) = \frac{n_i}{n+s}, \quad 1 \leq i \leq d \quad (5.21)$$

$$P^+(x_i|\mathbf{n}) = \frac{n_i+s}{n+s}, \quad 1 \leq i \leq d \quad (5.22)$$

Note how these bounds depend on \mathbf{n} and the additive parameter s but not on d (the RIP is thus satisfied). These lower and upper bounds can be transformed into a belief function because *bel* can be interpreted as a lower bound where the remaining mass is assigned to the frame of discernment.

$$m[\mathbf{n}](x_i) = \frac{n_i}{n+s}, \quad 1 \leq i \leq d \quad (5.23)$$

$$m[\mathbf{n}](\Theta) = \frac{s}{n+s} \quad (5.24)$$

The hyper-parameter s is usually an integer with $s \geq 1$. In [Walley, 1996], the author suggests $s = 1$ or $s = 2$. In the remainder of this chapter, $s = 1$ is used.

5.4.4. Belief Maximization

The approach proposed in [Dencœux, 2006] is based on computing simultaneous confidence intervals for the true probabilities and then deriving a belief function from these confidence intervals by solving a linear optimization problem. The resulting belief function must satisfy two requirements. The first requirement is derived from Hacking’s frequency principle [Hacking, 1965], which states that the degree of belief in an event should be equal to the probability of that event as the number of observations goes to infinity.

$$\lim_{n \rightarrow \infty} bel[\mathbf{n}](X) = P^*(X), \quad \forall X \subseteq \Theta \quad (5.25)$$

The second requirement is that the belief function should be less committed than or equal to the true probability function P^* at some confidence level $1 - \alpha$ for a finite random sample.

$$P(bel[\mathbf{n}](X) \leq P^*(X)) \geq 1 - \alpha, \quad \forall X \subseteq \Theta \quad (5.26)$$

The constant α is usually a small positive value (the author suggests $\alpha = 0.05$, though greater values tend to work better for the classification problem considered in this chapter as shown in Sect. 5.4.6).

Simultaneous confidence intervals for the true probability values are constructed which corresponds to a lower bound P^- and an upper bound P^+ on

the parameters of P^* at confidence level $1 - \alpha$. The confidence intervals are defined as follows [Goodman, 1965]:

$$P^-(x_i|\mathbf{n}) = \frac{a + 2n_i - \sqrt{\Delta_i}}{2(n + a)}, \quad (5.27)$$

$$P^+(x_i|\mathbf{n}) = \frac{a + 2n_i + \sqrt{\Delta_i}}{2(n + a)}, \quad (5.28)$$

$$\Delta_k = a(a + \frac{4n_i(n - n_i)}{n}). \quad (5.29)$$

Here, a is the quantile of order $1 - \alpha$ of the χ^2 -distribution with one degree of freedom.

The lower bound P^- satisfies both of the above-stated requirements. However, it cannot generally be transformed into a belief function. Instead, the set $\mathcal{B}(P^-)$ containing all belief functions satisfying $bel(X) \leq P^-(X|\mathbf{n}), \forall X \subseteq \Theta$ is considered. Within this set, the “most committed” belief function is chosen (e.g., the vacuous belief function would not be useful). As a measure of committedness, the author proposes to use the sum of all belief values. Maximizing this measure corresponds to the linear optimization problem

$$m^*[\mathbf{n}] = \arg \max_{m[\mathbf{n}] \in \mathcal{B}(P^-)} \sum_{X \subseteq \Theta} bel[\mathbf{n}](X) \quad (5.30)$$

subject to these three constraints:

$$\sum_{Y \subseteq X} m[\mathbf{n}](Y) \leq P^-(X|\mathbf{n}), \quad \forall X \subset \Theta, \quad (5.31)$$

$$\sum_{X \subseteq \Theta} m[\mathbf{n}](X) = 1, \quad (5.32)$$

$$m[\mathbf{n}](X) \geq 0, \quad \forall X \subseteq \Theta. \quad (5.33)$$

Constraint (5.31) asserts that the solution satisfies requirement (5.26) while constraints (5.32) and (5.33) assert that the solution is a valid mass function. Under these constraints, the optimal solution to (5.30) is shown to also satisfy requirement (5.25) in [Dencœux, 2006]. The solution can be obtained using basically any linear optimization algorithm.⁶

The problem of the optimization approach is its computational complexity because the number of constraints grows exponentially with d , thus making optimization infeasible even for moderately-sized spaces. For this reason, an approximate solution is presented for the case where a “meaningful ordering” of the elements in Θ can be defined (e.g., for a discretized real-valued variable). Assuming there exists a total ordering such that $\forall i, k : i < k \Rightarrow x_i < x_k$, the approximation consists of only allowing sequences $X_{i:k} = \{x_i, \dots, x_k\}$ with

⁶In the *PyDS* implementation, SciPy’s “COBYLA” optimization algorithm is used.

$i \leq k$ as focal sets. This restriction reduces number of potential focal sets from $2^d - 1$ to $d(d+1)/2$. Furthermore, it allows the solution to the maximum belief criterion (5.30) to be constructed analytically.

$$m[\mathbf{n}](X_{i:k}) = \begin{cases} P_{i:k}^- & \text{if } i = k, \\ P_{i:k}^- - P_{i+1:k}^- - P_{i:k-1}^- & \text{if } i + 1 = k, \\ P_{i:k}^- - P_{i+1:k}^- - P_{i:k-1}^- + P_{i+1:k-1}^- & \text{if } i + 1 < k. \end{cases} \quad (5.34)$$

$$P_{i:k}^- = P^-(X_{i:k}|\mathbf{n}) \quad (5.35)$$

5.4.5. MCD

The approach proposed in [Aregui and Denœux, 2008] is similar to the previous one in that it also makes use of Goodman's confidence intervals. It is based on finding the “most-committed dominating” (MCD) belief function from the set of belief functions induced by the confidence intervals. Instead of applying Hacking's frequency principle though, a weaker requirement for the resulting belief function more in line with the TBM and its distinction between a credal level and a pignistic level (see Sect. 1.1) is formulated: only the pignistic transformation of the belief function must be equal to P^* for $n \rightarrow \infty$. Note that the approach can be applied to both discrete and real-valued random variables, however, only the discrete case is considered here.

First, the confidence intervals defined by (5.27) and (5.28) are used to construct the set $S(\mathbf{n})$ of probability functions containing P^* at some confidence level $1 - \alpha$. Like in the previous approach, the authors suggest a small value for α (e.g., $\alpha = 0.05$) but larger values tend to work significantly better for classification.

$$S(\mathbf{n}) = \{P' \in \mathbb{P} | P^-(x_i|\mathbf{n}) \leq P'(x_i) \leq P^+(x_i|\mathbf{n}), 1 \leq i \leq d\} \quad (5.36)$$

$$P(P^* \in S(\mathbf{n})) \geq 1 - \alpha \quad (5.37)$$

Each element in $S(\mathbf{n})$ represents a pignistic probability distribution. The pignistic transformation of the sought belief function is contained in $S(\mathbf{n})$ at confidence level $1 - \alpha$.

Because a pignistic probability function P and the set of corresponding *isopignistic* belief functions (whose pignistic transformations are equal to P) are in a one-to-many correspondence, an additional criterion is needed for uniquely recovering a belief function from P . This criterion is the *least commitment principle* [Smets, 1993]. It states that, given a set of admissible belief functions, one should always select the least committed one.⁷ Here, committedness is defined in terms of q -ordering [Dubois et al., 2008]. A mass function m_1 is said to be q -more committed than a mass function m_2 (written as $m_1 \sqsubseteq_q m_2$) if $q_1(X) \leq q_2(X), \forall X \subseteq \Theta$.

⁷The least commitment principle is conceptually similar to the maximum entropy principle in probability theory.

Given a probability function $P \in \mathbb{P}$, the q -least committed isopignistic belief function m with $BetP_m = P$ is unique and can be defined in terms of a possibility distribution [Dubois et al., 2008, Dubois and Prade, 1983]

$$poss(x_i) = \sum_{k=1}^d \min(P(x_i), P(x_k)). \quad (5.38)$$

The corresponding belief function is consonant and therefore has only up to $d+1$ focal sets. Assuming the elements of Θ are arranged such that $poss(x_1) \geq poss(x_2) \geq \dots \geq poss(x_d)$, the belief function can be recovered from the possibility function in the following way [Dubois and Prade, 1982]:

$$m(X) = \begin{cases} 1 - poss(x_1) & \text{if } X = \emptyset, \\ poss(x_i) - poss(x_{i+1}) & \text{if } X = X_{1:i} \text{ with } i < d, \\ poss(x_d) & \text{if } X = \Theta, \\ 0 & \text{otherwise.} \end{cases} \quad (5.39)$$

Let $\mathcal{B}(\mathbf{n}) = \{m | BetP_m = P, P \in S(\mathbf{n})\}$ be the set of consonant belief functions whose pignistic transformations are in $S(\mathbf{n})$. Requiring elements in $\mathcal{B}(\mathbf{n})$ to be consonant is a simplification that makes the subsequent optimization tractable. Usually, the set $\mathcal{B}(\mathbf{n})$ does not contain a unique least-committed belief function. This is why the subset of belief functions $\mathcal{B}^*(\mathbf{n}) = \{m | m \in \mathcal{B}(\mathbf{n}), m' \sqsubseteq_q m, \forall m' \in \mathcal{B}(\mathbf{n})\}$ dominating all other belief functions in $\mathcal{B}(\mathbf{n})$ in terms of their q -ordering is considered. The authors then propose to select the *most committed* belief function from the set $\mathcal{B}^*(\mathbf{n})$. Because of the restriction to consonant belief functions, this most committed dominating belief function is unique (the proof can be found in [Aregui and Dencœux, 2008]) and can be obtained via its corresponding possibility function $poss^*$ with

$$poss^*(x_i) = \sup_{P \in S(\mathbf{n})} poss_P(x_i), \quad \forall x_i \in \Theta \quad (5.40)$$

where $poss_P$ is computed according to Eq. (5.38). The possibility function $poss^*$ can be computed by solving the following linear maximization problems:

$$poss^*(x_i) = \max_{P \in S(\mathbf{n})} \sum_{k=1}^d \min(P(x_i), P(x_k)), \quad 1 \leq i \leq d. \quad (5.41)$$

Like in the approach in [Dencœux, 2006], the number of constraints grows exponentially with the dimensionality d , which is why the optimization quickly becomes intractable. However, there exists a simple upper bound \widetilde{poss}^* , which can be used as an approximation.

$$\widetilde{poss}^*(x_i) = \min \left(1, \sum_{k=1}^d \min(P^+(x_i|\mathbf{n}), P^+(x_k|\mathbf{n})) \right) \quad (5.42)$$

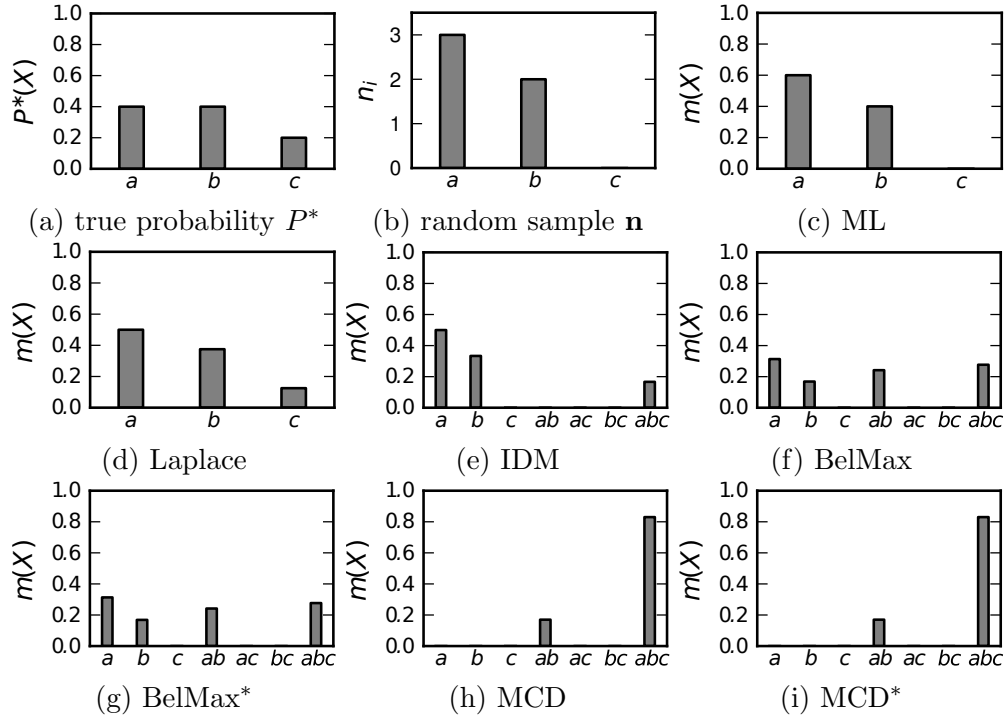


Figure 5.5.: Belief function construction example for $n = 5$ and frame of discernment $\Theta = \{a, b, c\}$. Fig. (a) represents the unknown probability distribution P^* while Fig. (b) represents a randomly drawn sample from P^* . Fig. (c) to (i) represent the distributions resulting from applying the different belief function constructing approaches to the observed random sample \mathbf{n} .

Finally, the approximate solution \tilde{m}^* of the MCD isopignistic belief function can be obtained from \widetilde{poss}^* using Eq. (5.39). Note that if one is only interested in the plausibility values of singletons (which is the case for classification, see below), then the transformation into a mass function can be omitted because the plausibility values are directly provided by the possibility distribution.

5.4.6. Comparison

Fig. 5.5 shows an example of the belief/probability functions resulting from applying the different approaches to a small random sample. The frame of discernment contains only three elements in this example with $\Theta = \{a, b, c\}$, making it possible to show all focal sets. The true distribution P^* is defined by the unknown parameter vector $(0.4, 0.4, 0.2)$ and the observed sample counts \mathbf{n} are $(3, 2, 0)$. The maximum likelihood solution (ML) is trivial because the observed relative frequencies are directly accepted. Laplace smoothing comes quite close to the true distribution in this example, in particular, because the true probability of c is greater than 0. The Imprecise Dirichlet Model (IDM) solution

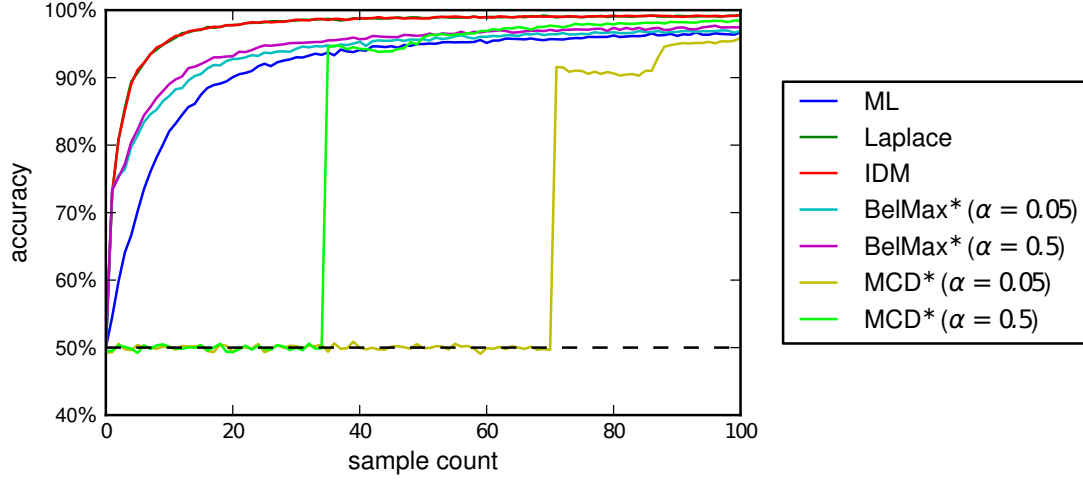


Figure 5.6.: Mean classification accuracy in relation to the sample count resulting from different belief construction methods on a synthetic dataset. The dashed line indicates the chance level.

assigns a mass value of roughly 0.2 to Θ , making the small sample count explicit while preserving the observed ratio between a and b . The approach based on maximizing the sum of belief values (BelMax) assigns even more mass to Θ and also some to the union $\{a, b\}$. In this example, the approximate solution BelMax* where the singletons are assumed to possess a natural order is identical to the solution obtained via optimization. The MCD solutions (both the optimization version and the approximation denoted by MCD*) result in highly non-committed belief functions, which is caused by the small sample count and the fact that only the pignistic transformation is required to be equal to the true distribution for $n \rightarrow \infty$. Both for the BelMax and the MCD solutions, $\alpha = 0.5$ is used for the confidence level. By comparison, the MCD solutions are vacuous when using the recommended value $\alpha = 0.05$. This high degree of ignorance turns out to be problematic for classification.

Before comparing the different approaches in the context of an object recognition task, they are first compared in a more controlled classification setting using synthetic data. In this setting, there are 2 classes and 10 features with 10 possible values each. The true joint distribution $P(x, z_{1:10})$ of classes and features is assumed to follow a naive Bayes model with conditionally independent features.

$$P(x, z_{1:10}) \propto P(x) \prod_{i=1}^{10} P(z_i|x) \quad (5.43)$$

The prior $P(x)$ and the likelihoods $P(z_i|x)$ are categorical distributions with random parameters.⁸ In order to account for the fact that feature distributions

⁸Each parameter is assigned a random number in the interval $[0, 1)$, after which normalization is performed.

for real-world classification problems tend to be rather non-uniform, an entropy criterion is used where a randomly generated feature distribution is only accepted if its entropy is less than 80% of the maximum possible entropy. From each of the resulting feature distributions $P(z_i|x)$, several samples are drawn based on which mass functions are constructed using the presented methods. Afterwards, classification is performed as described in Sect. 5.3.1 (the prior distribution is ignored and chosen to be vacuous for the evidential methods and uniform for the Bayesian methods). For each construction method and each sample count (ranging from 0 to 100), this process is repeated 20,000 times and the mean classification accuracy is calculated.⁹

Fig. 5.6 shows the resulting accuracies in relation to the number of samples used for constructing each feature distribution (i.e., the total number of samples for constructing the entire model is ten times the sample count for each distribution). As expected, for a sample count of 0, all methods perform at chance level while a sample count of 100 provides the highest accuracy. Because the frame of discernment for each feature consists of 10 elements, computing exact solutions for the BelMax and MCD approaches is intractable. Instead, only the approximate solutions BelMax* and MCD* are considered. For the confidence level parameter α , both $\alpha = 0.5$ and the suggested smaller value $\alpha = 0.05$ are tested. As seen, the higher value consistently outperforms the smaller one, both for BelMax* and MCD*. The MCD* method only performs at chance level for small sample counts and then instantaneously jumps to an accuracy level of 0.9 or higher. The reason is that, for small sample counts, the method does not assign any mass to singletons, even if the random sample strongly favors one (small values for α make the effect worse). This can also be seen in Fig. 5.5 where no mass is assigned to singletons by the MCD solutions.

Apart from this “defect” of the MCD method for small sample counts, the maximum likelihood method performs worst, even for the largest sample count of 100. The BelMax method consistently outperforms it, in particular for small sample counts. The best performing methods are Laplace smoothing and IDM with practically identical accuracies, which can in part be explained by their conceptual similarity. These results are somewhat surprising because the simplest methods show the best performance in this setting, one of them being a probabilistic method. Though not the best-performing methods, the BelMax and MCD approaches are promising in that they consistently outperform the maximum likelihood approach.

An interesting direction for future research on belief function construction methods would be to consider specialized methods for generative classification models. This is because all the presented methods construct full mass functions over the feature space whereas only plausibilities of singletons are actually used during classification (apart from the prior, which only plays a minor role). As

⁹Accuracy is defined as the number of correctly classified items divided by the total number of items.

a result, it is not necessary to consider the full power set over the feature space (with an exponential number of parameters in the worst case) and strong restrictions like the assumption of consonance could be avoided because, for singleton plausibilities, the number of parameters is only equal to the cardinality of the feature space.

5.5. Application to Object Recognition

In this section, the recognition architecture presented in Sect. 5.3 is applied to an object recognition problem. Inspired by the approach in [Schill et al., 2001], the system presented in this section performs recognition by iteratively processing regions of interest in an image. This resembles the way humans analyze scenes via saccadic eye movements where the information gain strategy can be interpreted as a model for attention. For each region of interest, a descriptor is extracted which is used in conjunction with the performed action to update the belief distribution over possible object classes. In this case, an action simply corresponds to a position in the image. For a physical implementation of the recognition architecture where actions correspond to 3D movements of a camera, see [Kluth et al., 2013]. Note that the aim of this section is to demonstrate how the recognition architecture works in an application context and not to compete with current state-of-the-art object recognition approaches (e.g., [Zeiler and Fergus, 2013]).

5.5.1. Dataset

The dataset used in this section is Caltech-256 [Griffin et al., 2007], which is a standard dataset for object recognition containing 256 object classes. In order to limit the computational effort, only a subset of 10 randomly selected classes is used (the only criterion being that each selected class contains at least 100 images). For each class, 100 images are randomly selected where 80 are used for the training set and 20 for the test set. With the number of images being the same for each class, the class prior can be ignored during inference (i.e., it is assumed to be vacuous or uniform). Furthermore, mean classification accuracies are not skewed by the overrepresentation of some classes.

All images used here are RGB color images. Before processing an image, it is scaled to a size of 256×256 pixels using bicubic interpolation. The aspect ratio is preserved during scaling by cropping the image to a quadratic size afterwards. In Fig. 5.7, mean images are shown for each of the selected classes. These give an impression of how variable the spatial layout is for each class and they show the importance of positional information provided by actions.

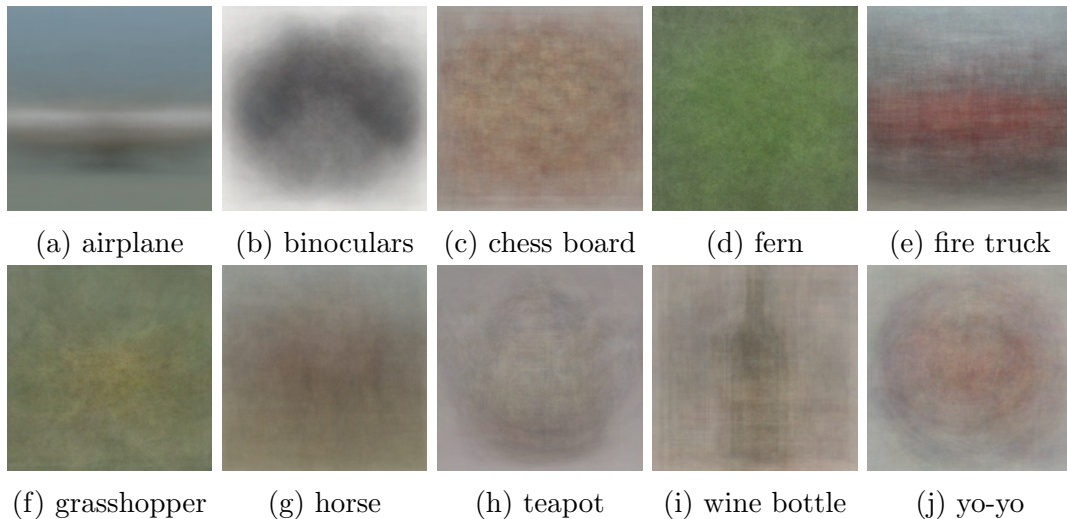


Figure 5.7.: Mean class images from the Caltech-256 dataset for 10 selected classes. Each mean image is constructed by averaging over all images from the corresponding class where each image is first scaled and cropped to a size of 256×256 pixels.

5.5.2. Visual Processing

A region of interest corresponds to a 64×64 pixel patch. In order to extract a feature vector from a region of interest, the *gist of a scene* is used [Oliva and Torralba, 2001, Oliva, 2005]. While originally intended as a global scene descriptor, it also works well as a local region of interest descriptor. A gist feature vector essentially represents a histogram of the outputs of orientation-selective bandpass filters. For the implementation, the open source library *pyleargist* is used.¹⁰

The main reason why gist features are chosen is that they perform best in terms of classification accuracy. Used in the recognition system presented here, they consistently outperform other popular descriptors like local binary patterns [Ojala et al., 1996] and Haralick’s descriptor [Haralick et al., 1973] (see [Wolter et al., 2009] for a comparison of gist features to other texture descriptors). For future work, it would be interesting though to also consider local keypoint descriptors like SIFT [Lowe, 2004] and SURF [Bay et al., 2008], in particular because these have been shown to be very successful for object recognition [Csurka et al., 2004].

Because inference is based on belief functions over the sensory space (see Sect. 5.3.1), gist vectors are discretized using vector quantization in order to obtain a discrete representation. In this case, k-means clustering with $k = 100$ is performed using randomly initialized centroids. For learning the prototype vectors, 20,000 gist vectors are extracted from randomly sampled image patches.

¹⁰<https://pypi.python.org/pypi/pyleargist>

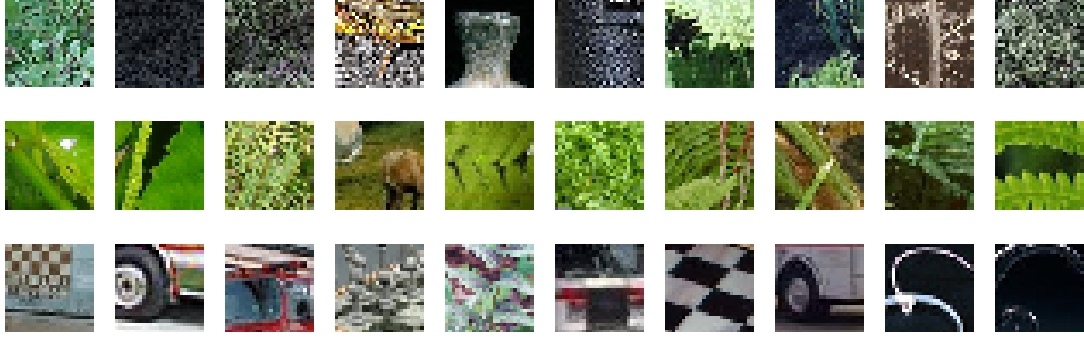


Figure 5.8.: Image patches belonging to three selected clusters of gist vectors. Each row corresponds to a cluster, i.e., each patch within a row results in a gist vector assigned to the same cluster.

Using these prototypes, each region of interest can simply be described by the corresponding prototype index where $|\Theta_z| = 100$. While the primary reason for this vector quantization approach is the ability to estimate belief distributions over the feature space, it is a common approach for object recognition in the context of bag-of-feature models [Csurka et al., 2004].

Fig. 5.8 shows examples of image patches whose corresponding gist vectors are assigned to the same prototype. Some prototypes appear to represent quite meaningful features, for example, the patches in the first row all contain textures with high image frequencies while the patches in the second row all contain various forms of plants. Usually, patches represented by the same prototype are not as homogeneous though and the third row contains very heterogeneous structures like chess boards and car tires. Clearly, such heterogeneities cause ambiguities regarding recognition. While larger values for k reduce these ambiguities, there is a trade-off between the prototype quality and the number of parameters that have to be estimated (see next section). The choice $k = 100$ seems to result in a good trade-off between under- and overfitting of the resulting model.

5.5.3. Sensorimotor Model

As described in Sect. 5.3.1, inference is performed based on the class-conditional plausibility functions $pl_{\Theta_z}[x, u_t]$. Like the sensory information z_t , actions are discretized using k-means clustering (with $k = 50$). An action u_t is a pair of image coordinates representing the center of a region of interest. For the vector quantization, 20,000 image positions are randomly sampled, resulting in a prototype distribution that is approximately uniform over an image.¹¹

The basis for constructing $pl_{\Theta_z}[x, u_t]$ is a sensorimotor histogram where, for

¹¹Alternatively, image coordinates could be discretized using binning but the vector quantization approach is more general and it resembles the processing of visual information more closely.

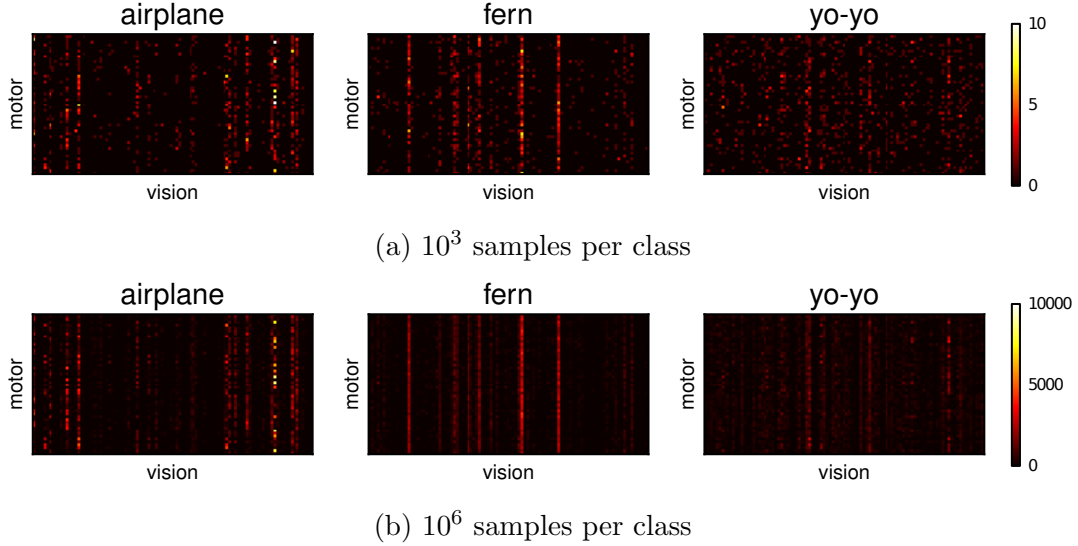


Figure 5.9.: Sensorimotor feature histograms for three selected classes from the Caltech-256 dataset. Each pixel in the six shown images represents an entry in the corresponding histogram where action prototypes u_t are shown on the vertical axis and sensory prototypes z_t are shown on the horizontal axis. In (a), each histogram is generated using 10^3 samples while the histograms in (b) are constructed from 10^6 samples.

each combination of a class x and a quantized action u_t , the number of occurrences of each quantized gist vector z_t is counted. By applying one of the belief construction methods presented in Sect. 5.4, a mass function is then obtained and from that the singleton plausibilities. Examples of sensorimotor histograms for three classes are shown in Fig. 5.9. The effect of small sample counts is clearly visible in these histograms where the histograms generated using 10^3 samples are much noisier than the histograms generated from 10^6 samples (a sample is a tuple (u_t, z_t) randomly drawn from the training set). Note that during training, samples are drawn at random image positions while, during recognition, features are only extracted at the centroids of actions.

Generally, some classes exhibit much sparser feature distributions than others. For example, class “yo-yo” is not very sparse, which is reflected by the low accuracy for this class (see Sect. 5.5.5). Also visible is that the correlation of actions and sensory information is stronger for some classes, e.g., for class “airplane”, some sensory features are much more frequent for certain actions than for others. In contrast, for a class like “fern”, this correlation is much weaker as indicated by the almost continuous vertical lines for 10^6 samples.

A sensorimotor feature is modeled as a pair of a quantized region-of-interest-descriptor and a quantized absolute position here. An alternative would be to consider sensorimotor features consisting of two subsequent sensory features and

an action representing the relative movement between these two features (this is how sensorimotor features are defined in [Schill et al., 2001]). In fact, such “triplet” features were tested but the resulting accuracy was generally lower, presumably because of the significantly higher number of parameters that have to be estimated in this case.

5.5.4. Example

In order to give an impression of how the system presented in this chapter works in practice, Fig. 5.10 shows an example run. Here, the system analyzes an image belonging to the class “binoculars” and correctly classifies it after having performed a number of actions. The underlying belief model is constructed using the IDM method.

First, the expected information gain is computed for all of the 50 possible actions and the action with the maximum expected gain is executed. This is shown in Fig. 5.10a with the information gain distribution superimposed over the image. A gist vector is then extracted from the image patch indicated by the white square which is used, together with the motor information, to update the initially vacuous belief distribution. This process continues until, after 14 performed actions (shown in Fig. 5.10f), the belief reaches a confidence threshold (a pignistic probability of at least 0.99) and the recognition is terminated. Note that once an action has been performed, it cannot be performed a second time and the expected information gain is 0 for the corresponding position.

The pignistic transformation of the belief distribution over time is plotted in Fig. 5.10g. Here, it can be seen that other classes are initially more likely and only after 6 actions, the true class emerges as the most likely one.

5.5.5. Results

This section presents a systematic evaluation of the object recognition system on the Caltech-256 dataset. All results shown below are obtained using 10-fold cross validation. In all cases, a vacuous prior is used (a uniform one for the Bayesian approaches).

First, the different belief construction methods presented in Sect. 5.4 are compared based on their classification accuracy in relation to the number of samples used for constructing each model. The sample count determines how many sensorimotor features are used to compute the sensorimotor histogram for each class. The results of this comparison are shown in Fig. 5.11. Aside from some variations for very low sample counts, all methods yield higher accuracies for larger sample counts, though there appears to be a limit at around 60% where additional samples do not improve accuracy. Overall, the IDM method performs best with Laplace smoothing being a close second. Unlike in the accuracy comparison on the synthetic dataset shown in Fig. 5.6, the BelMax* method performs worse than the ML solution. The MCD* method (with $\alpha = 0.5$)

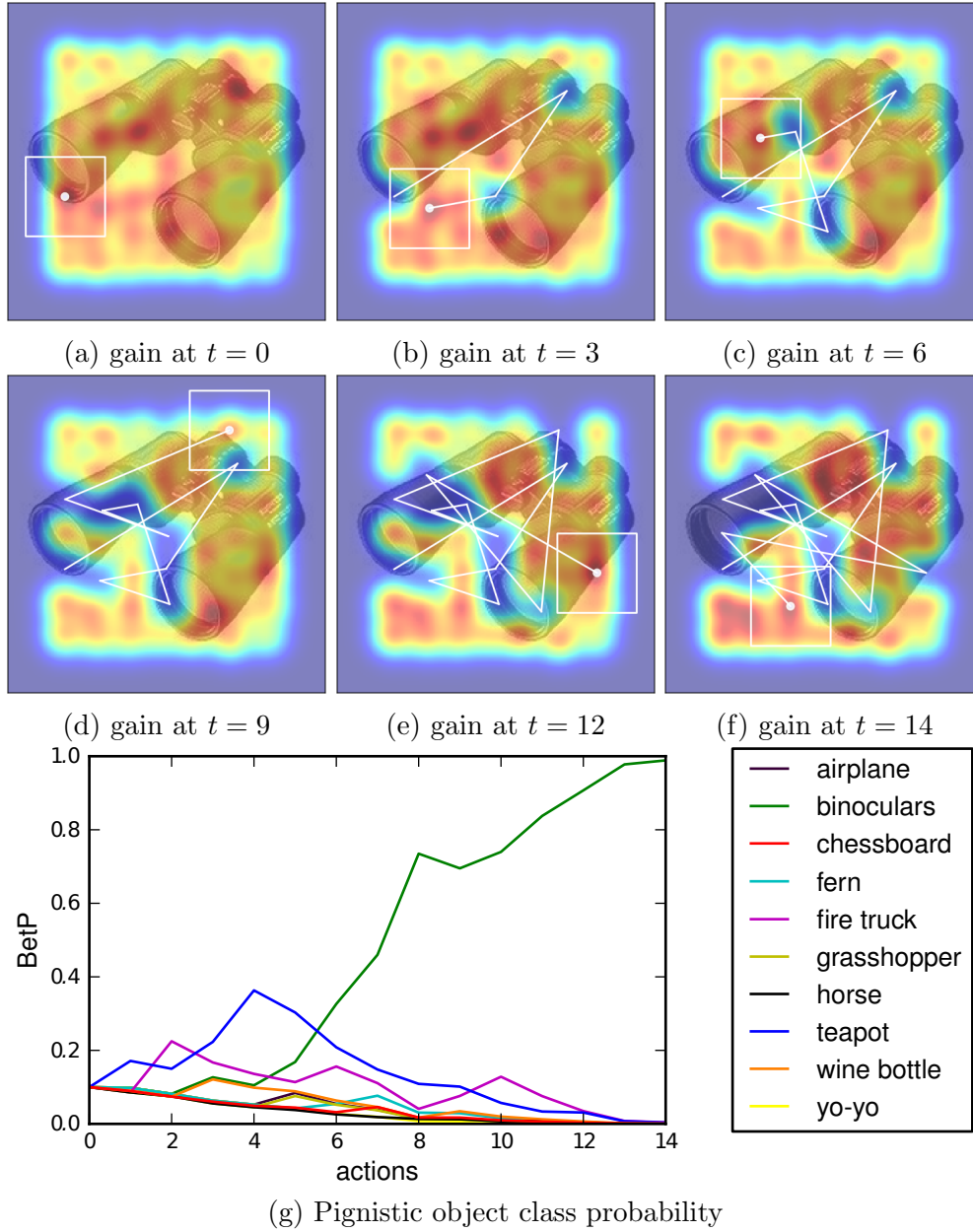


Figure 5.10.: Object recognition example. Plots (a) to (f) show the performed actions and the information gain distribution for potential target position at different points in time. The white rectangle represents the current fixation while the white line indicates the sequence of fixated positions over time. The information gain is superimposed using a Gaussian located at each prototype position to interpolate between positions where red colors indicate high expected information gain values and blue colors indicate lower values. Plot (g) shows the pignistic object class probability over time. The true class “binoculars” is correctly recognized with high confidence after the system has performed 14 actions.

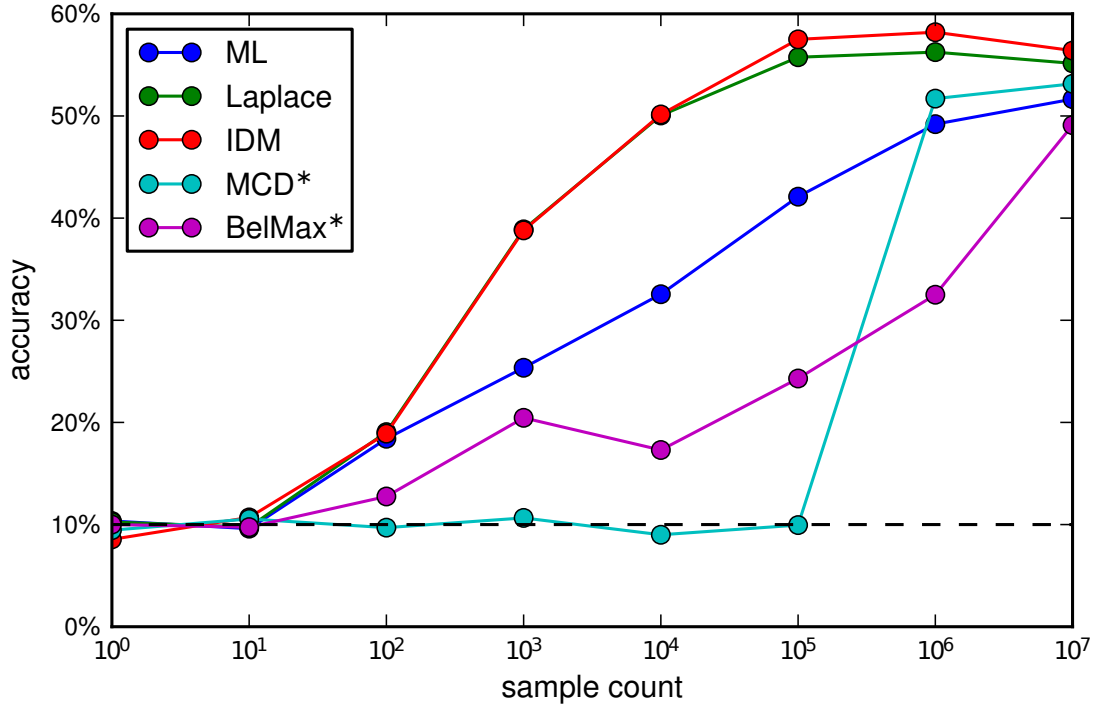


Figure 5.11.: Mean classification accuracy on Caltech-256 (using 10 classes) for different model learning methods and different sample counts. The sample count is plotted on a logarithmic scale. Chance level is indicated by the dashed line.

only performs at chance level until reaching 10^6 samples, at which point it outperforms the ML solution. This effect is the same seen for the synthetic dataset where about 35 samples are required in order to obtain better-than-chance accuracy (the number of histogram entries is much larger here, thus explaining why the sample count needs to be significantly higher).

Because there are only 80 training images for each class, for a sample count of 10^7 , this means 125,000 sensorimotor features are extracted from each image. Considering that there are only $(256 - 64)^2 = 36,864$ possible image positions, the resulting information is very redundant. This is why the visible accuracy limit is not actually an inherent property of the model but rather the result of a limited amount of data. It also explains why some methods (IDM and Laplace) do not appear to improve with additional samples. In contrast, a method like BelMax* greatly improves with this additional, mostly-redundant information, showing that the method does not optimally utilize the already available information for smaller sample counts (choosing even greater values for α might reduce this problem).

Fig. 5.12 shows the classification accuracy broken down by classes for a sample count of 10^7 . Not surprisingly, some classes are generally easier to recognize than others, for example, the accuracy for class “airplane” is much higher than

IDM	91.0	62.0	47.5	66.5	69.0	45.0	49.5	53.5	54.0	26.0	56.4
Laplace	90.5	63.0	46.0	68.0	70.0	40.0	45.5	50.5	52.5	25.5	55.1
MCD*	69.5	51.5	53.5	56.5	67.0	44.5	56.0	41.0	50.0	42.0	53.2
BelMax*	89.0	35.5	70.0	52.5	73.0	26.0	52.0	32.0	46.0	15.0	49.1
ML	83.5	59.0	40.0	66.5	65.0	38.5	39.0	44.5	50.5	30.0	51.6
no motor	92.0	66.5	38.5	67.0	67.0	28.0	49.0	35.5	59.0	20.5	52.3
inf. gain	90.5	62.0	48.5	68.0	72.0	42.0	50.0	51.0	55.5	26.5	56.6
training	98.8	94.6	89.0	91.4	94.6	88.8	88.0	94.5	92.5	86.0	91.8
NN	95.5	21.0	66.0	85.0	24.0	28.0	32.0	45.5	35.0	9.5	44.1
	airplane	binoculars	chess board	fern	fire truck	grasshopper	horse	teapot	wine bottle	yo-yo	all

Figure 5.12.: Mean classification accuracy broken down by classes using a sample count of 10^7 (the column on the right shows results averaged across classes). The first 5 rows correspond to the results shown in Fig. 5.11. The bottom 4 rows show additional results where “no motor” means the action information u_t is omitted, “inf. gain” means information gain is used during recognition (opposed to random behavior), “training” means classification is performed on the training set, and “NN” means a nearest-neighbor classifier is used instead of belief function inference.

for class “yo-yo” regardless of the model construction method. This can be explained by the fact that images from the “airplane” class tend to be quite similar (in particular regarding perspective) while images from the “yo-yo” class tend to be very heterogeneous. What is noticeable though is that there are significant differences between the belief construction methods within single classes. For example, while the BelMax* method performs worst when averaged across all classes, it is actually the best-performing method for some classes (“fire truck” and “chess board”, for the latter by a wide margin). Another interesting effect is that the MCD* method performs worst among all methods for what appears to be the easiest class (“airplane”) but significantly outperforms all other methods for the most difficult class (“yo-yo”). The MCD* method generally appears to be the most “robust” method with no class accuracy below 41%, which can be explained by its high degree of non-committedness.

For comparison, Fig. 5.12 shows some additional results obtained using the IDM method. A lack of motor information only appears to negatively affect the accuracy for some classes (e.g., “chess board”, “grass hopper”, and “teapot”) while for other classes, it makes no difference or even leads to improved perfor-

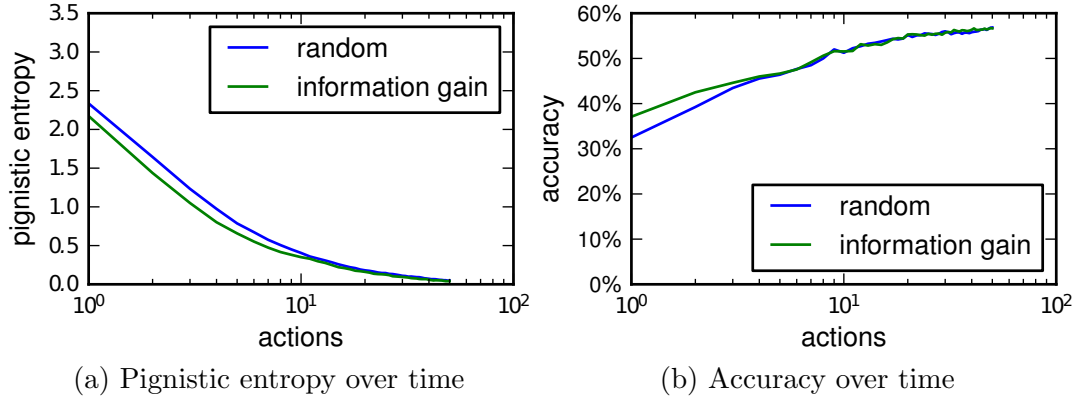


Figure 5.13.: Comparison of information gain maximization with random behavior. Plot (a) shows the pignistic entropy H_{BetP} at each point in time while (b) shows the corresponding classification accuracy, both averaged over all test images using cross-validation. The action count/time is shown on a logarithmic scale to better visualize the effect of the information gain strategy for the first few actions.

mance (overfitting is reduced in this case because the number of model parameters is lower). As expected, the use of information gain (opposed to random behavior, based on which all other results are generated) makes no difference regarding classification accuracy because it only influences the order in which evidence is collected. The most interesting result is perhaps the very high accuracy on the training set, which means that overfitting is a severe problem and that more images should be used for model construction (ideally, training and test set accuracy should converge). As a reference and to show that belief function inference is a valid approach for classification, accuracy resulting from a simple nearest neighbor classifier (NN) is also shown. Here, the local gist feature vectors from all prototype positions are concatenated in a fixed order to construct a global feature vector. The resulting mean accuracy of 44.1% is significantly lower than for belief function inference though.

Finally, Fig. 5.13 shows the effect of the information gain strategy on uncertainty and accuracy over time in comparison to random action selection. As expected, the result of maximizing the expected information gain is that the uncertainty measured by the pignistic entropy decreases more quickly (see Fig. 5.13a). The fact that the difference in uncertainty is rather small can in part be explained by the noisiness of the data and the loss of information caused by quantization (e.g., with car tires and chess board features mapped to the same prototype as shown in Fig. 5.8, the prediction of the next sensory feature is quite uncertain). In particular during the first few actions though, the uncertainty is consistently lower and the corresponding accuracy higher (see Fig. 5.13b). Uncertainty and accuracy converge eventually (accuracy more quickly than uncertainty) and they are in fact identical at $t = 50$ because the

final belief distribution is independent of the evidence gathering order.

5.6. Discussion

The architecture presented in this chapter uses belief function theory for inference and an information gain maximization strategy for actively and efficiently collecting evidence over time. The architecture itself is essentially domain-independent and can therefore be applied to a wide variety of problems. For the problem of object recognition, it was shown that a belief function model can outperform an equivalent probabilistic model in terms of classification accuracy. In addition, it was shown how belief functions can be obtained from data and that the additional dimension of uncertainty provided by belief functions can be used to reduce the problem of small sample counts during training.

In the comparison of the different model construction methods, the IDM approach resulted in the highest accuracy. However, it was revealed that a simple probabilistic approach like Laplace smoothing can yield almost equally good results. As argued in Sect. 5.4.6, the problem of deriving belief functions from data remains an open research question and approaches tailored to generative classification models would be an interesting subject of future work.

Regarding the information gain strategy, it was shown that it outperforms random behavior in terms of how quickly uncertainty decreases and accuracy increases. However, there is a trade-off between the relatively high computational effort of maximizing the expected information gain and the resources that are saved by selecting goal-directed actions. This trade-off always depends on the application and the costs of performing an action (e.g., the costs of a diagnostic test in a medical domain could be quite severe). A general way of improving this trade-off would be to lower the computational complexity of the information gain computation by relying on approximations (e.g., by sampling actions instead of considering all possibilities). Note that the information gain strategy presented here is greedy because it only considers the next action. For applications like active localization, where multiple actions are required to reach certain locations, such a greedy approach could fail and action sequences would have to be considered [Reineking, 2008]. Finally, one interesting question for future work would be to investigate optimization criteria other than the expected information gain. While the information gain causes the uncertainty to reduce as quickly as possible, an alternative criterion could be based on maximizing the expected accuracy.

6

Conclusion

6.1. Summary

In this thesis, belief function theory was applied to problems involving temporal uncertainty, spatial uncertainty, and uncertainty during decision making. In particular, it was shown how Bayesian solutions can be generalized in order to allow for the use of richer uncertainty models based on belief functions. The main contributions in each area are both theoretical and algorithmic with a focus on exploitation of independence and efficient Monte-Carlo approximations.

Evidential Particle Filtering

Chap. 3 showed how filtering can be performed within the belief function framework. For this, evidential filtering equations were derived and they were shown to be generalizations of Bayesian filtering. These equations make it possible to recursively estimate the current state of a dynamical system if the state prior, observations, and/or state transitions are described by belief functions. However, the computational complexity of each update turns out to be exponential with $O(2^{|\Theta|})$ where $|\Theta|$ denotes the size of the finite state space.

In order to reduce this complexity, a Monte-Carlo algorithm was proposed which approximates the belief distribution of the current state using a finite number of samples. This Monte-Carlo algorithm constitutes a generalization of discrete probabilistic particle filters. The use of sampling reduces the computational complexity from exponential to linear with $O(K|\Theta|)$ where K denotes the number samples. By using this approximation, filtering becomes feasible in much larger state spaces while the approximation error tends to be limited if the underlying mass functions have a quasi-sparse structure where only a

limited number of focal sets have mass values significantly larger than 0. In the correction step of the algorithm, importance sampling is performed for efficiently incorporating new observations. This importance sampling approach is not limited to filtering though because it provides a general solution to approximating Dempster's combination rule, in particular if one of the belief functions is derived from the generalized Bayesian theorem. In fact, the inference in the recognition architecture presented in Chap. 5 is based on the same algorithm.

Evidential SLAM

In Chap. 4, an evidential solution to the SLAM problem was proposed, which is one of the most fundamental problems in mobile robotics. During SLAM, a mobile robot has to construct a map of an unknown environment while localizing itself based on this map. The proposed algorithm creates evidential occupancy grid maps where the state of each cell is described by a belief function. This provides the robot with additional information because a lack of evidence can be distinguished from conflicting measurements. Previous approaches for evidential mapping focused exclusively on the mapping aspect of SLAM and did not consider the joint estimation problem characteristic of SLAM. The proposed approach uses a Rao-Blackwellized particle filter to approximate the joint distribution of the robot's path and the map, and thereby generalizes the popular FastSLAM algorithm. Rao-Blackwellization means that the joint distribution is factorized into a path estimation problem and a conditional mapping problem where the map, conditioned on a particular path, can be updated analytically. The validity of this factorization is proved in Sect. A.2. The factorization is necessary because, for an occupancy grid map consisting of M cells (usually with $M > 10,000$), there are 2^M possible maps and 2^{2^M} possible focal sets, thus making it impossible to cover the joint space with sufficiently many particles. By additionally assuming that grid cells are independent of each other, the computational complexity of incorporating a new measurement becomes linear in M with $O(KM)$ where K denotes the number particles.

In addition to the evidential FastSLAM algorithm, forward and inverse sonar sensor models based on belief functions were presented. Evidential inverse sensor models, which provide a local map based on a single measurement, are often defined in a heuristic manner while evidential forward models, which specify the plausibility of a measurement given a map, are not considered at all. In contrast, here, an evidential forward sensor model was presented and the inverse sensor model was directly derived from it. Both forward and inverse sensor models can be evaluated in linear time with $O(M)$.

Active Evidential Recognition

In Chap. 5, a recognition architecture was presented that actively selects evidence. In this context, two main problems need to be solved: inference and

action selection. The inference is based on a generative model where features are assumed to be conditionally independent. By adopting the Monte-Carlo algorithm from the correction step of the evidential particle filter, incorporating a new observation can be done with time complexity $O(K |\Theta|)$ instead of $O(2^{|\Theta|})$ ($|\Theta|$ denotes the number of classes and K denotes the number of samples). Actions are selected based on the principle of maximum expected information gain. In order to compute the expected information gain of an action, the pignistic probability distribution of the next feature needs to be computed. Computing this distribution analytically is intractable because of the exponential complexity $O(2^{|\Theta_z| |\Theta|})$ where $|\Theta_z|$ denotes the size of the feature space. However, by applying a sampling-based approximation paired with an additional prior assumption, the complexity is reduced to $O(K |\Theta_z| |\Theta|)$ (the number of samples K can be different for inference and information gain computation).

The architecture itself is domain-independent and has been applied in other context before. In this thesis, it was applied to an object recognition task. Belief functions were used to cope with the problem that the amount of training data is often not sufficient to reliably estimate model parameters. Different methods from the literature for constructing belief functions from small amounts of data were compared empirically and it was shown that an evidential approach was able to outperform a corresponding Bayesian approach regarding recognition rate. In addition, it was shown that the information gain strategy causes uncertainty to decrease more quickly than for random action selection.

6.2. Outlook

With respect to the applications presented in this thesis, possible extensions and improvements have already been pointed out in the respective chapters. Therefore, the outlook provided here is more of a global one.

On the theoretical side, one area of belief function theory that is not fully developed yet is that of belief functions for continuous domains. While there are multiple works on this subject, some of the key tools of the TBM framework in particular have not been generalized to continuous domains. For example, continuous observations can be handled by the generalized Bayesian theorem as discussed in [Smets, 2005a] but no solution exists for continuous states. The same applies to the disjunctive rule of combination which is essential for constructing conditional belief functions [Smets, 1993]. Under which conditions these tools can be extended to continuous domains remains an open question for future research. In the absence of these tools for continuous domains, one alternative solution, aside from discretization, is the use of hybrid models where the continuous part of a problem is modeled probabilistically while belief functions are used for modeling the discrete aspects. An example of such a hybrid approach is the evidential FastSLAM algorithm presented in Chap. 4 where the marginal path distribution is a probability density function and the discrete

map is modeled as a belief function.

Another area of belief function theory that should be explored further is that of uncertainty measures. While an active area of research for quite some time [Klir, 2004], there is still no consensus on appropriate measures for quantifying the different dimensions of belief function uncertainty. As shown in Chap. 5, pignistic entropy works in practice as a measure of total uncertainty but it also disregards all the “interesting information” (i.e., the non-probabilistic information) due to the pignistic transformation. Besides the particular problem of identifying suitable measures of uncertainty, an even more ambitious goal would be the development of a generalized information theory for belief functions.

Regarding algorithms for belief function theory, there has been significant progress for coping with the exponential complexity of representing and combining belief functions. Decompositions based on independence assumptions paired with Monte-Carlo approximations effectively reduce complexity and make many problems tractable as shown in this thesis. One problem where new approaches could have a significant impact is belief function construction from data [Aregui and Denœux, 2008]. While some of the approaches that were compared empirically in this thesis showed promising results, none appear to constitute a definitive solution to this very essential problem.

Perhaps the most fundamental task for future research is the development of more convincing applications of belief function theory. This concerns the modeling/inference aspect as well as the problem of decision making based on belief functions. An example of such an application could be the evidential SLAM algorithm presented in Chap. 4. The value of belief functions in this application could become even more apparent if actions performed by the robot would directly reflect the additional information provided by an evidential map. To this date, belief function theory still has somewhat of an “outsider status” compared to Bayesian approaches. The best way of changing this and making belief functions a widely-accepted tool is therefore to develop more applications that undeniably demonstrate the value provided by belief function theory.

Appendices



Proofs

A.1. Belief-Probability Combination

Let m_1 and m_2 be mass functions defined over the frame of discernment Θ . Let m_1 be Bayesian with $m_1(a) = P(a)$, $\forall a \in \Theta$. Then the following holds for the combination $m_1 \oplus m_2$ for all $A \subseteq \Theta$:

$$(m_1 \oplus m_2)(A) = \begin{cases} \eta P(A) pl_2(A) & \text{if } |A| = 1, \\ 0 & \text{else,} \end{cases} \quad (\text{A.1})$$

$$\eta^{-1} = \sum_{b \in \Theta} P(b) pl_2(b). \quad (\text{A.2})$$

Proof

The mass function $m_1 \oplus m_2$ is Bayesian because all its focal sets are subsets of the focal sets of m_1 and m_2 . Because m_1 is Bayesian, all its focal sets have cardinality 1, which is why the same is true for the combined mass function.

Therefore, only singletons $a \in \Theta$ have to be considered:

$$\begin{aligned} & (m_1 \oplus m_2)(a) \\ &= \frac{\sum_{B \cap C = \{a\}} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)} \quad (\text{Dempster's rule}) \end{aligned} \quad (\text{A.3})$$

$$= \frac{\sum_{\{b\} \cap C = \{a\}} m_1(b) m_2(C)}{1 - \sum_{\{b\} \cap C = \emptyset} m_1(b) m_2(C)} \quad (|B| > 1 \Rightarrow m_1(B) = 0) \quad (\text{A.4})$$

$$= \frac{\sum_{\{a\} \cap C = \{a\}} m_1(a) m_2(C)}{1 - \sum_{\{b\} \cap C = \emptyset} m_1(b) m_2(C)} \quad (\{b\} \cap C = \{a\} \Rightarrow b = a) \quad (\text{A.5})$$

$$= \frac{\sum_{\{a\} \cap C = \{a\}} P(a) m_2(C)}{1 - \sum_{\{b\} \cap C = \emptyset} P(b) m_2(C)} \quad (m_1(a) = P(a)) \quad (\text{A.6})$$

$$= \frac{P(a) \sum_{C \ni a} m_2(C)}{1 - \sum_{\{b\} \cap C = \emptyset} P(b) m_2(C)} \quad (a \text{ is fixed}) \quad (\text{A.7})$$

$$= \frac{P(a) pl_2(a)}{1 - \sum_{\{b\} \cap C = \emptyset} P(b) m_2(C)} \quad \left(\sum_{C \ni a} m_2(C) = pl_2(a) \right) \quad (\text{A.8})$$

$$= \frac{P(a) pl_2(a)}{1 - \sum_b P(b) \sum_{C \not\ni b} m_2(C)} \quad (\text{split sum}) \quad (\text{A.9})$$

$$= \frac{P(a) pl_2(a)}{1 - \sum_b P(b) bel_2(\bar{b})} \quad \left(\sum_{C \not\ni b} m_2(C) = bel_2(\bar{b}) \right) \quad (\text{A.10})$$

$$= \frac{P(a) pl_2(a)}{1 - \sum_b P(b) + \sum_b P(b) pl_2(b)} \quad (bel_2(\bar{b}) = 1 - pl_2(b)) \quad (\text{A.11})$$

$$= \frac{P(a) pl_2(a)}{\sum_b P(b) pl_2(b)} \quad \left(\sum_b P(b) = 1 \right) \quad (\text{A.12})$$

■

A.2. Belief-Probability Product Rule

Let m be a mass function defined over the product space $\Theta \times \Omega$. Let the marginal distribution of m over Ω be Bayesian, i.e., $m^{\downarrow \Omega}(b) = P(b)$, $\forall b \in \Omega$. Then the following factorization holds:

$$m(A, b) = m[b](A) P(b), \quad \forall A \subseteq \Theta, b \in \Omega. \quad (\text{A.13})$$

In case $P(b) = 0$, the product $m[b](A) P(b)$ is defined to be 0 as well.

Proof

(Remark: Subscripts are used to indicate over which space each particular function is defined.)

$$P_{\Omega}(b) m_{\Theta}[b](A) \quad (\text{A.14})$$

$$= P_{\Omega}(b) \left(m_{\Theta \times \Omega} \oplus m_{\Omega; b}^{\uparrow \Theta \times \Omega} \right)^{\downarrow \Theta} (A) \quad (\text{cond. with } m_{\Omega; b}(b) = 1) \quad (\text{A.15})$$

$$= P_{\Omega}(b) \sum_{B \subseteq \Omega} (m_{\Theta \times \Omega} \oplus m_{\Omega; b}^{\uparrow \Theta \times \Omega})(A, B) \quad (\text{marginalization}) \quad (\text{A.16})$$

$$= P_{\Omega}(b) (m_{\Theta \times \Omega} \oplus m_{\Omega; b}^{\uparrow \Theta \times \Omega})(A, b) \quad (m_{\Omega; b}^{\uparrow \Theta \times \Omega}(\Theta, b) = 1) \quad (\text{A.17})$$

$$= P_{\Omega}(b) \frac{\sum_{(A', B') \cap (\Theta, b) = (A, b)} m_{\Theta \times \Omega}(A', B')}{1 - \sum_{(A', B') \cap (\Theta, b) = \emptyset} m_{\Theta \times \Omega}(A', B')} \quad (\text{Dempster's rule}) \quad (\text{A.18})$$

$$= P_{\Omega}(b) \frac{\sum_{B' \ni b} m_{\Theta \times \Omega}(A, B')}{1 - \sum_{(A', B') \cap (\Theta, b) = \emptyset} m_{\Theta \times \Omega}(A', B')} \quad (A' \cap \Theta = A \Rightarrow A' = A) \quad (\text{A.19})$$

$$= P_{\Omega}(b) \frac{m_{\Theta \times \Omega}(A, b)}{1 - \sum_{(A', B') \cap (\Theta, b) = \emptyset} m_{\Theta \times \Omega}(A', B')} \quad (|B| > 1 \Rightarrow m_{\Theta \times \Omega}^{\downarrow \Omega}(B) = 0) \quad (\text{A.20})$$

$$= P_{\Omega}(b) \frac{m_{\Theta \times \Omega}(A, b)}{1 - \sum_{A' \subseteq \Theta} \sum_{B' \not\ni b} m_{\Theta \times \Omega}(A', B')} \quad (\text{split sum}) \quad (\text{A.21})$$

$$= P_{\Omega}(b) \frac{m_{\Theta \times \Omega}(A, b)}{1 - \sum_{B' \not\ni b} m_{\Theta \times \Omega}^{\downarrow \Omega}(B')} \quad (\text{marginalization}) \quad (\text{A.22})$$

$$= P_{\Omega}(b) \frac{m_{\Theta \times \Omega}(A, b)}{1 - \sum_{b' \neq b} P_{\Omega}(b')} \quad (m_{\Theta \times \Omega}^{\downarrow \Omega}(b') = P(b')) \quad (\text{A.23})$$

$$= m_{\Theta \times \Omega}(A, b) \quad (1 - \sum_{b' \neq b} P_{\Omega}(b') = P_{\Omega}(b)) \quad (\text{A.24})$$

■

B

Software

Two open-source libraries for belief function theory have been developed in the context of this thesis. The first is called *JDS* (Java Dempster-Shafer library) and is written in Java.¹ The second is called *PyDS* (Python Dempster-Shafer library) and is a Python implementation.² Both are licensed under the GPL. *PyDS* is essentially the successor to *JDS* and provides more advanced features.

B.1. PyDS

The following list outlines the most important features provided by the PyDS library.

Efficient representation Efficient storage and processing of quasi-sparse mass functions based on hash tables.

Combination rules Various rules for combining belief functions (e.g., conjunctive/Dempster’s rule, disjunctive, cautious, etc.). In addition, efficient Monte-Carlo algorithms for most combination rules based on importance sampling.

Conversion between representations Conversion between the most common belief representations m , bel , pl , and q .

Optional normalization Support for both normalized and unnormalized belief functions.

¹<http://sourceforge.net/projects/jds>

²<https://github.com/reineking/pyds>

Generalized Bayesian theorem Generalized Bayesian theorem for different belief representations including efficient Monte-Carlo approximations.

Particle filtering Prediction and correction step implementations for filtering as presented in Chapter 3 (with support for analytical and sampling-based inference).

Pignistic transformation Pignistic transformation for decision making.

Uncertainty measures Uncertainty measures like pignistic entropy and local conflict [Pal et al., 1993].

Belief functions from data Implementations for all the belief construction methods presented in Sect. 5.4.

Own Publications

- [Kluth et al., 2013] Kluth, T., Nakath, D., Reineking, T., Zetsche, C., and Schill, K. (2013). Sensorimotor integration using an information gain strategy in application to object recognition tasks. *Perception*, 42:223. ECVF Abstract.
- [Reineking, 2008] Reineking, T. (2008). Active vision-based localization using Dempster-Shafer theory. Master’s thesis, University of Bremen, Department of Informatics.
- [Reineking, 2011] Reineking, T. (2011). Particle filtering in the Dempster-Shafer theory. *International Journal of Approximate Reasoning*, 52(8):1124–1135.
- [Reineking and Clemens, 2013] Reineking, T. and Clemens, J. (2013). Evidential FastSLAM for grid mapping. In *16th International Conference on Information Fusion*, pages 789–796.
- [Reineking et al., 2008] Reineking, T., Kohlhagen, C., and Zetsche, C. (2008). Efficient wayfinding in hierarchically regionalized spatial environments. In Freksa, C., Newcombe, N., Gärdenfors, P., and Wöfl, S., editors, *Spatial Cognition VI*, volume 5248 of *Lecture Notes in Computer Science*, pages 56–70. Springer Berlin/Heidelberg.
- [Reineking et al., 2009] Reineking, T., Schult, N., and Hois, J. (2009). Evidential combination of ontological and statistical information for active scene classification. In *KEOD*, pages 72–79. INSTICC Press.
- [Reineking et al., 2011] Reineking, T., Schult, N., and Hois, J. (2011). Combining statistical and symbolic reasoning for active scene categorization. In Fred, A., Dietz, J. L. G., Liu, K., and Filipe, J., editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management*, volume 128 of *Communications in Computer and Information Science*, pages 262–275. Springer Berlin/Heidelberg.
- [Reineking et al., 2010] Reineking, T., Wolter, J., Gadzicki, K., and Zetsche, C. (2010). Bio-inspired architecture for active sensorimotor localization. In Hölscher, C., Shipley, T., Olivetti Belardinelli, M., Bateman,

- J., and Newcombe, N., editors, *Spatial Cognition VII*, volume 6222 of *Lecture Notes in Computer Science*, pages 163–178, Portland, Oregon. Springer Berlin/Heidelberg.
- [Schult et al., 2013] Schult, N., Reineking, T., and Kluss, T. (2013). Information-driven audio-visual source localization on a mobile robot. In *International Conference on Multisensory Motor Behaviour: Impact of Sound*, pages 40–41. Abstract.
- [Wolter et al., 2009] Wolter, J., Reineking, T., Zetsche, C., and Schill, K. (2009). From visual perception to place. *Cognitive Processing*, 10:351–354. Extended Abstract.
- [Zetsche et al., 2008] Zetsche, C., Reineking, T., Wolter, J., and Schill, K. (2008). Active vision for exploratory localization. *Journal of Vision*, 8(6):1152. Abstract.

Bibliography

- [Aregui and Denœux, 2008] Aregui, A. and Denœux, T. (2008). Constructing consonant belief functions from sample data using confidence sets of pignistic probabilities. *International Journal of Approximate Reasoning*, 49(3):575–594.
- [Bar-Shalom et al., 2004] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.
- [Barnett, 1981] Barnett, J. A. (1981). Computational methods for a mathematical theory of evidence. In *Proceedings of the 7th international joint conference on Artificial intelligence*, pages 868–875. Morgan Kaufmann Publishers Inc.
- [Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359.
- [Ben Yaghlane et al., 2002a] Ben Yaghlane, B., Smets, P., and Mellouli, K. (2002a). Belief function independence: I. the marginal case. *International Journal of Approximate Reasoning*, 29(1):47–70.
- [Ben Yaghlane et al., 2002b] Ben Yaghlane, B., Smets, P., and Mellouli, K. (2002b). Belief function independence: Ii. the conditional case. *International Journal of Approximate Reasoning*, 31(1):31–75.
- [Carlson et al., 2005] Carlson, J., Murphy, R. R., Christopher, S., and Casper, J. (2005). Conflict metric as a measure of sensing quality. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2032–2039.
- [Csurka et al., 2004] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.
- [Delmotte and Smets, 2004] Delmotte, F. and Smets, P. (2004). Target identification based on the transferable belief model interpretation of Dempster-Shafer model. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 34(4):457–471.

- [Dempster, 1966] Dempster, A. P. (1966). New methods for reasoning towards posterior distributions based on sample data. *The Annals of Mathematical Statistics*, 37(2):355–374.
- [Dempster, 1967] Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, 38(2):325–339.
- [Dempster, 1968] Dempster, A. P. (1968). A generalization of Bayesian inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 205–247.
- [Dempster, 2001] Dempster, A. P. (2001). Normal belief functions and the Kalman filter. *Data Analysis from Statistical Foundations*, pages 65–84.
- [Denœux, 1995] Denœux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics*, 25(5):804–813.
- [Denœux, 2000] Denœux, T. (2000). A neural network classifier based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 30(2):131–150.
- [Denœux, 2006] Denœux, T. (2006). Constructing belief functions from sample data using multinomial confidence regions. *International Journal of Approximate Reasoning*, 42(3):228–252.
- [Denœux, 2008] Denœux, T. (2008). Conjunctive and disjunctive combination of belief functions induced by non-distinct bodies of evidence. *Artificial Intelligence*, 172(2-3):234–264.
- [Denœux and Masson, 2012] Denœux, T. and Masson, M.-H. (2012). Evidential reasoning in large partially ordered sets. *Annals of Operations Research*, 195(1):135–161.
- [Denœux and Smets, 2006] Denœux, T. and Smets, P. (2006). Classification using belief functions: Relationship between case-based and model-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36(6):1395–1406.
- [Doucet et al., 2001] Doucet, A., De Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo methods in practice*. Springer New York.
- [Doucet et al., 2000] Doucet, A., De Freitas, N., Murphy, K., and Russell, S. (2000). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183.

- [Dubois and Prade, 1982] Dubois, D. and Prade, H. (1982). On several representations of an uncertain body of evidence. *Fuzzy information and decision processes*, pages 167–181.
- [Dubois and Prade, 1983] Dubois, D. and Prade, H. (1983). Unfair coins and necessity measures: towards a possibilistic interpretation of histograms. *Fuzzy sets and systems*, 10(1):15–20.
- [Dubois and Prade, 1986a] Dubois, D. and Prade, H. (1986a). On the unicity of Dempster’s rule of combination. *International Journal of Intelligent Systems*, 1(2):133–142.
- [Dubois and Prade, 1986b] Dubois, D. and Prade, H. (1986b). A set-theoretic view of belief functions – Logical operations and approximations by fuzzy sets. *International Journal of General Systems*, 12(3):193–226.
- [Dubois et al., 2008] Dubois, D., Prade, H., and Smets, P. (2008). A definition of subjective possibility. *International Journal of Approximate Reasoning*, 48(2):352–364.
- [Duong et al., 2005] Duong, T. V., Bui, H. H., Phung, D. Q., and Venkatesh, S. (2005). Activity recognition and abnormality detection with the switching hidden semi-Markov model. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 838–845. IEEE.
- [Durrant-Whyte and Bailey, 2006] Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part i. *IEEE Robotics & Automation Magazine*, 13(2):99–110.
- [Elfes, 1989] Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57.
- [Eliazar and Parr, 2003] Eliazar, A. and Parr, R. (2003). DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 1135–1142.
- [Ferson et al., 2003] Ferson, S., Kreinovich, V., Ginzburg, L., Myers, D. S., and Sentz, K. (2003). Constructing probability boxes and dempster-shafer structures. Technical report, Sandia National Laboratories.
- [Gambino et al., 1997] Gambino, F., Ulivi, G., and Vendittelli, M. (1997). The transferable belief model in ultrasonic map building. In *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, volume 1, pages 601–608. IEEE.
- [Goodman, 1965] Goodman, L. A. (1965). On simultaneous confidence intervals for multinomial proportions. *Technometrics*, 7(2):247–254.

- [Gordon and Shortliffe, 1985] Gordon, J. and Shortliffe, E. H. (1985). A method for managing evidential reasoning in a hierarchical hypothesis space. *Artificial Intelligence*, 26(3):323–357.
- [Gordon et al., 2002] Gordon, N. J., Maskell, S., and Kirubarajan, T. (2002). Efficient particle filters for joint tracking and classification. In *SPIE 4728, Signal and Data Processing of Small Targets*, pages 439–449.
- [Gordon et al., 1993] Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET.
- [Griffin et al., 2007] Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. Technical report, California Institute of Technology.
- [Hacking, 1965] Hacking, I. (1965). *Logic of statistical inference*. Cambridge University Press.
- [Hady et al., 2011] Hady, M. F. A., Schwenker, F., and Palm, G. (2011). Multi-view forest: A new ensemble method based on Dempster-Shafer evidence theory. *International Journal of Applied Mathematics and Statistics (IJAMAS): Special Issue on Soft Computing and Approximate Reasoning*, 22(S11):2–19.
- [Hähnel et al., 2003] Hähnel, D., Burgard, W., Fox, D., and Thrun, S. (2003). An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 206–211. IEEE.
- [Haralick et al., 1973] Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, (6):610–621.
- [Harmanec, 1997] Harmanec, D. (1997). *Uncertainty in Dempster-Shafer theory*. PhD thesis, Binghamton, NY, USA.
- [Hartley, 1928] Hartley, R. V. L. (1928). Transmission of information. *The Bell Labs Technical Journal*, 7(3).
- [Isard and Blake, 1998] Isard, M. and Blake, A. (1998). CONDENSATION—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28.
- [Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

- [Keynes, 1921] Keynes, J. M. (1921). *A Treatise on Probability*, chapter The Principle of Indifference, pages 41–64. Macmillan and Co.
- [Khaleghi et al., 2013] Khaleghi, B., Khamis, A., Kararay, F. O., and Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44.
- [Klein et al., 2010] Klein, J., Lecomte, C., and Miché, P. (2010). Hierarchical and conditional combination of belief functions induced by visual tracking. *International Journal of Approximate Reasoning*, 51(4):410–428.
- [Klir, 1999] Klir, G. J. (1999). Uncertainty and information measures for imprecise probabilities: An overview. In *Proceedings of the First International Symposium on Imprecise Probabilities and Their Applications*.
- [Klir, 2004] Klir, G. J. (2004). Generalized information theory: aims, results, and open problems. *Reliability Engineering & System Safety*, 85(1–3):21–38.
- [Klir, 2005] Klir, G. J. (2005). *Uncertainty and information: foundations of generalized information theory*. Wiley.
- [Klir and Smith, 2001] Klir, G. J. and Smith, R. M. (2001). On measuring uncertainty and uncertainty-based information: Recent developments. *Annals of Mathematics and Artificial Intelligence*, 32(1–4):5–33.
- [Kohlas and Monney, 1994] Kohlas, J. and Monney, P. A. (1994). Theory of evidence – a survey of its mathematical foundations, applications and computational aspects. *Zeitschrift für Operations Research*, 39(1):35–68.
- [Kohlas and Monney, 2008] Kohlas, J. and Monney, P. A. (2008). An algebraic theory for statistical information based on the theory of hints. *International Journal of Approximate Reasoning*, 48(2):378–398.
- [Kolmogorov, 1933] Kolmogorov, A. N. (1933). *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer, Berlin.
- [Kreinovich et al., 1994] Kreinovich, V., Bernat, A., Borrett, W., Mariscal, Y., and Villa, E. (1994). *Monte-Carlo methods make Dempster-Shafer formalism feasible*, pages 175–191. Wiley.
- [Kuncheva et al., 2001] Kuncheva, L. I., Bezdek, J. C., and Duin, R. P. W. (2001). Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314.
- [Kurdej et al., 2012] Kurdej, M., Moras, J., Cherfaoui, V., and Bonnifait, P. (2012). Map-aided fusion using evidential grids for mobile perception in urban environment. *Proceedings of the 2nd International Conference on Belief Functions*, 164:343–350.

- [La Scala and Morelande, 2008] La Scala, B. and Morelande, M. (2008). An analysis of the single sensor bearings-only tracking problem. In *11th International Conference on Information Fusion*, pages 1–6. IEEE.
- [Lefevre et al., 2002] Lefevre, E., Colot, O., and Vannoorenberghe, P. (2002). Belief function combination and conflict management. *Information fusion*, 3(2):149–162.
- [Li et al., 2007] Li, X., Huang, X., Dezert, J., Duan, L., and Wang, M. (2007). A successful application of DSMT in sonar grid map building and comparison with DST-based approach. *International Journal of Innovative Computing, Information and Control*, 3(3):539–549.
- [Liu, 1996] Liu, L. (1996). A theory of Gaussian belief functions. *International Journal of Approximate Reasoning*, 14(2):95–126.
- [Ljung, 1979] Ljung, L. (1979). Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24(1):36–50.
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.
- [Montemerlo et al., 2002] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National conference on Artificial Intelligence*, pages 593–598.
- [Montemerlo et al., 2003] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *International Joint Conference on Artificial Intelligence*, 18:1151–1156.
- [Moral and Salmerón, 1999] Moral, S. and Salmerón, A. (1999). A Monte-Carlo algorithm for combining Dempster-Shafer belief based on approximate pre-computation. *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pages 305–315.
- [Moral and Wilson, 1996] Moral, S. and Wilson, N. (1996). Importance sampling Monte-Carlo algorithms for the calculation of Dempster-Shafer belief. *Proceeding of IPMU*, 96.
- [Moras et al., 2011] Moras, J., Cherfaoui, V., and Bonnifait, P. (2011). Credibilist occupancy grids for vehicle perception in dynamic environments. In *International Conference on Robotics and Automation (ICRA)*, pages 84–89.
- [Moravec, 1988] Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74.

- [Mullane et al., 2006] Mullane, J., Adams, M. D., and Wijesoma, W. S. (2006). Evidential versus bayesian estimation for radar map building. In *ICARCV'06. 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–8. IEEE.
- [Muñoz-Salinas et al., 2009] Muñoz-Salinas, R., Medina-Carnicer, R., Madrid-Cuevas, F. J., and Carmona-Poyato, A. (2009). Multi-camera people tracking using evidential filters. *International Journal of Approximate Reasoning*, 50(5):732–749.
- [Noë, 2004] Noë, A. (2004). *Action in Perception*. MIT Press.
- [Ojala et al., 1996] Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59.
- [Oliva, 2005] Oliva, A. (2005). Gist of a scene. *Neurobiology of Attention*, pages 251–256.
- [Oliva and Torralba, 2001] Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175.
- [O’Regan and Noë, 2001] O’Regan, K. and Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, 24:939–973.
- [Orponen, 1990] Orponen, P. (1990). Dempster’s rule of combination is #P-complete. *Artificial Intelligence*, 44(1-2):245–253.
- [Pagac et al., 1998] Pagac, D., Nebot, E. M., and Durrant-Whyte, H. (1998). An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 14(4):623–629.
- [Pal et al., 1992] Pal, N., Bezdek, J., and Hemasinha, R. (1992). Uncertainty measures for evidential reasoning I: A review. *International Journal of Approximate Reasoning*, 7(1):165–183.
- [Pal et al., 1993] Pal, N., Bezdek, J., and Hemasinha, R. (1993). Uncertainty measures for evidential reasoning II: A new measure of total uncertainty. *International Journal of Approximate Reasoning*, 8(1):1–16.
- [Palm, 2012] Palm, G. (2012). *Novelty, information and surprise*. Springer.
- [Pawlak, 1982] Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356.

- [Pichon and Denœux, 2010] Pichon, F. and Denœux, T. (2010). The unnormlized Dempster’s rule of combination: A new justification from the least commitment principle and some extensions. *Journal of Automated Reasoning*, 45(1):61–87.
- [Prinz, 1990] Prinz, W. (1990). *A common coding approach to perception and action*, pages 167–203. Springer-Verlag, Berlin, relationships between perception and action: current approaches edition.
- [Quost et al., 2011] Quost, B., Masson, M.-H., and Denœux, T. (2011). Classifier fusion in the Dempster-Shafer framework using optimized t-norm based combination rules. *International Journal of Approximate Reasoning*, 52(3):353–374.
- [Ramasso et al., 2007a] Ramasso, E., Rombaut, M., and Pellerin, D. (2007a). Forward-backward-Viterbi procedures in the Transferable Belief Model for state sequence analysis using belief functions. *Lecture Notes Artificial Intelligence*, 4724:405–417.
- [Ramasso et al., 2007b] Ramasso, E., Rombaut, M., and Pellerin, D. (2007b). State filtering and change detection using tbm conflict application to human action recognition in athletics videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(7):944–949.
- [Ribo and Pinz, 2001] Ribo, M. and Pinz, A. (2001). A comparison of three uncertainty calculi for building sonar-based occupancy grids. *Robotics and Autonomous Systems*, 35(3-4):201–209.
- [Ristic et al., 2004] Ristic, B., Gordon, N., and Bessell, A. (2004). On target classification using kinematic data. *Information Fusion*, 5(1):15–21.
- [Russell et al., 2008] Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). LabelMe: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173.
- [Schill, 1995] Schill, K. (1995). Analysing uncertain data in decision support systems. In *Proceedings of ISUMA*, pages 437–442.
- [Schill, 1997] Schill, K. (1997). *Decision Support Systems with Adaptive Reasoning Strategies*, volume 1337 of *Lecture Notes in Computer Science: Foundations of Computer Science*, pages 417–427. Springer-Verlag, Berlin/Heidelberg.
- [Schill et al., 1991] Schill, K., Pöppel, E., and Zetsche, C. (1991). Completing knowledge by competing hierarchies. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, pages 348–352. Morgan Kaufmann Publishers Inc.

- [Schill et al., 2001] Schill, K., Umkehrer, E., Beinlich, S., Krieger, G., and Zetzsche, C. (2001). Scene analysis with saccadic eye movements: Top-down and bottom-up modeling. *Journal of Electronic Imaging*, 10(1):152–160.
- [Schill et al., 2009] Schill, K., Zetzsche, C., and Hois, J. (2009). A belief-based architecture for scene analysis: From sensorimotor features to knowledge and ontology. *Fuzzy Sets and Systems*, 160(10):1507–1516.
- [Sentz and Ferson, 2002] Sentz, K. and Ferson, S. (2002). Combination of evidence in Dempster-Shafer theory. Technical report, Sandia National Laboratories.
- [Shafer, 1976] Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ.
- [Shafer et al., 1987] Shafer, G., Shenoy, P. P., and Mellouli, K. (1987). Propagating belief functions in qualitative Markov trees. *International Journal of Approximate Reasoning*, 1(4):349–400.
- [Shafer and Tversky, 1985] Shafer, G. and Tversky, A. (1985). Languages and designs for probability judgment. *Cognitive Science*, 9(3):309–339.
- [Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27:379–423.
- [Shenoy and Shafer, 1990] Shenoy, P. P. and Shafer, G. (1990). Axioms for probability and belief-function propagation. In *Uncertainty in Artificial Intelligence*, pages 169–198. Morgan Kaufmann.
- [Smets, 1988] Smets, P. (1988). Belief functions. In Smets, P., Mamdani, E. H., Dubois, D., and Prade, H., editors, *Non Standard Logics for Automated Reasoning*, pages 253–286. Academic Press, London.
- [Smets, 1990] Smets, P. (1990). The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):447–458.
- [Smets, 1991] Smets, P. (1991). About updating. In D’ambrosio, B., Smets, P., and and, B. P. P., editors, *Uncertainty in Artificial Intelligence 91*, pages 378–385. Morgan Kaufmann, San Mateo, Ca, USA.
- [Smets, 1992a] Smets, P. (1992a). The concept of distinct evidence. In *Information Processing and Management of Uncertainty*, pages 89–94.
- [Smets, 1992b] Smets, P. (1992b). The nature of the unnormalized beliefs encountered in the transferable belief model. In *Proceedings of the Eighth international conference on Uncertainty in artificial intelligence*, pages 292–297.

- [Smets, 1992c] Smets, P. (1992c). Resolving misunderstandings about belief functions: A response to the many criticisms raised by Judea Pearl. *International Journal of Approximate Reasoning*, 6:321–344.
- [Smets, 1993] Smets, P. (1993). Belief functions: The disjunctive rule of combination and the generalized Bayesian theorem. *International Journal of Approximate Reasoning*, 9:1–35.
- [Smets, 1995] Smets, P. (1995). The canonical decomposition of a weighted belief. In *Int. Joint Conf. on Artificial Intelligence*, pages 1896–1901.
- [Smets, 1997] Smets, P. (1997). The α -junctions: combination operators applicable to belief functions. In Gabbay, D., Kruse, R., Nonnengart, A., and Ohlbach, H., editors, *Qualitative and quantitative practical reasoning*, pages 131–153. Springer.
- [Smets, 1998a] Smets, P. (1998a). The application of the transferable belief model to diagnostic problems. *International Journal of Intelligent Systems*, 13:127–157.
- [Smets, 1998b] Smets, P. (1998b). The transferable belief model for quantified belief representation. *Handbook of defeasible reasoning and uncertainty management systems*, 1:267–301.
- [Smets, 2000] Smets, P. (2000). Data fusion in the transferable belief model. In *Proceedings of the Third International Conference on Information Fusion*, pages 21–33.
- [Smets, 2002] Smets, P. (2002). Decision making in a context where uncertainty is represented by belief functions. In Srivastava, R. P. and Mock, T. J., editors, *Belief Functions in Business Decisions*, pages 17–61. Physica-Verlag, Heidelberg, Germany.
- [Smets, 2005a] Smets, P. (2005a). Belief functions on real numbers. *International Journal of Approximate Reasoning*, 40(3):181–223.
- [Smets, 2005b] Smets, P. (2005b). Decision making in the TBM: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning*, 38:133–147.
- [Smets, 2007] Smets, P. (2007). Analyzing the combination of conflicting belief functions. *Information Fusion*, 8(4):387–412.
- [Smets and Kennes, 1994] Smets, P. and Kennes, R. (1994). The transferable belief model. *Artificial Intelligence*, 66:191–234.

-
- [Smets and Ristic, 2004] Smets, P. and Ristic, B. (2004). Kalman filters and joint tracking and classification in the TBM framework. In *Proceedings of the Seventh International Conference on Information Fusion*, pages 46–53.
- [Smets and Ristic, 2007] Smets, P. and Ristic, B. (2007). Kalman filter and joint tracking and classification based on belief functions in the TBM framework. *Information Fusion*, 8(1):16–27.
- [Smith and Cheeseman, 1986] Smith, R. and Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56–68.
- [Smith et al., 1990] Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer.
- [Sutton and McCallum, 2010] Sutton, C. and McCallum, A. (2010). An introduction to conditional random fields. *arXiv preprint arXiv:1011.4088*.
- [Szczot et al., 2012] Szczot, M., Löhlein, O., and Palm, G. (2012). Dempster-Shafer fusion of context sources for pedestrian recognition. In Denceux, T. and Masson, M.-H., editors, *Belief Functions: Theory and Applications*, volume 164 of *Advances in Intelligent and Soft Computing*, pages 319–326. Springer Berlin Heidelberg.
- [Thoma, 1989] Thoma, H. M. (1989). *Factorization of belief functions*. PhD thesis, Cambridge, MA, USA.
- [Thrun, 2002] Thrun, S. (2002). Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, pages 1–35.
- [Thrun, 2003] Thrun, S. (2003). Learning occupancy grid maps with forward sensor models. *Autonomous robots*, 15(2):111–127.
- [Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT press Cambridge, MA.
- [Thrun et al., 2001] Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141.
- [Walley, 1996] Walley, P. (1996). Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society*, 58(1):3–57.
- [Walley, 2000] Walley, P. (2000). Towards a unified theory of imprecise probability. *International Journal of Approximate Reasoning*, 24(2):125–148.

- [Wan and van der Merwe, 2000] Wan, E. A. and van der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158.
- [Wilson, 2000] Wilson, N. (2000). *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 5, chapter Algorithms for Dempster-Shafer theory, pages 421–475. Springer.
- [Wilson and Moral, 1996] Wilson, N. and Moral, S. (1996). Fast Markov chain algorithms for calculating Dempster-Shafer belief. In *Proceedings of the 12th European Conference on Artificial Intelligence*, pages 672–676.
- [Xu and Smets, 1996] Xu, H. and Smets, P. (1996). Reasoning in evidential networks with conditional belief functions. *International Journal of Approximate Reasoning*, 14(2):155–185.
- [Yager, 1987] Yager, R. R. (1987). On the Dempster-Shafer framework and new combination rules. *Information Sciences*, 41(2):93–137.
- [Yager and Liu, 2008] Yager, R. R. and Liu, L. (2008). *Classic works of the Dempster-Shafer theory of belief functions*. Springer.
- [Yang and Aitken, 2006] Yang, T. and Aitken, V. (2006). Evidential mapping for mobile robots with range sensors. *IEEE Transactions on Instrumentation and Measurement*, 55(4):1422–1429.
- [Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3):338–353.
- [Zadeh, 1978] Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28.
- [Zadeh, 1979] Zadeh, L. A. (1979). On the validity of Dempster’s rule of combination of evidence. Technical report, Electronics Research Laboratory, University of California, Berkeley.
- [Zadeh, 1984] Zadeh, L. A. (1984). Review of a mathematical theory of evidence. *AI Magazine*, 5(3):81.
- [Zeiler and Fergus, 2013] Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional neural networks. *arXiv preprint arXiv:1311.2901*.
- [Zetsche et al., 1993] Zetsche, C., Barth, E., and Wegmann, B. (1993). *The Importance of Intrinsically Two-Dimensional Image Features in Biological Vision and Picture Coding*, pages 109–138. Digital images and human vision. MIT Press, Cambridge, MA.

- [Zetzsche et al., 2009] Zetzsche, C., Wolter, J., Galbraith, C., and Schill, K. (2009). Representation of space: image-like or sensorimotor. *Spatial Vision*, 22(5):409–424.
- [Zetzsche et al., 2008] Zetzsche, C., Wolter, J., and Schill, K. (2008). Sensorimotor representation and knowledge-based reasoning for spatial exploration and localisation. *Cognitive Processing*, 9:283–297.